Secure Incentives to Cooperate for Wireless Networks

THÈSE N° 3813 (2007)

Submitted Version

PRÉSENTÉE À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Naouel Ben Salem

Ingénieur Informaticienne Diplômée (M.Sc.)
Ecole Nationale des Sciences de l'informatique, Tunis, Tunisie

de nationalité tunisienne

acceptée sur proposition du jury:
Prof. Jean-Pierre Hubaux, directeur de thèse
Prof. Emre Telatar, président du jury
Prof. Markus Jakobsson, rapporteur
Prof. Patrick Thiran, rapporteur
Prof. Don Towsley, rapporteur
Lausanne, EPFL
2007

Contents

Abstract

The operating principle of certain wireless networks makes essential the cooperation between the mobile nodes. However, if each node is an autonomous selfish entity, cooperation is not guaranteed and therefore we need to use incentive techniques. In this thesis, we study cooperation in three different types of networks: WiFi networks, Wireless Mesh Networks (WMNs), and Hybrid Ad-hoc networks. Cooperation has a different goal for each of these networks, we thus propose incentive mechanisms adapted to each case.

In the first chapter of this thesis, we consider WiFi networks whose wide-scale adoption is impeded by two major hurdles: the lack of a seamless roaming scheme and the variable QoS experienced by the users. We devise a reputation-based solution that (i) allows a mobile node to connect to a foreign Wireless ISP in a secure way while preserving his anonymity and (ii) encourages the WISPs to cooperate, i.e., to provide the mobile clients with a good QoS. Cooperation appears here twofold: First, the mobile clients have to collaborate in order to build and maintain the reputation system and second, the use of this reputation system encourages the WISPs to cooperate. We show, by means of simulations, that our reputation model indeed encourages the WISPs to behave correctly and we analyze the robustness of our solution against various attacks.

In the second chapter of the thesis, we consider Wireless Mesh Networks (WMNs), a new and promising paradigm that uses multi-hop communications to extend WiFi networks. Indeed, by connecting only one hot spot to the Internet and by deploying several Transit Access Points (TAPs), a WISP can extend its coverage and serve a large number of clients at a very low cost. We analyze the characteristics of WMNs and deduce three fundamental network operations that need to be secured: (i) the routing protocol, (ii) the detection of corrupt TAPs and (iii) the enforcement of a proper fairness metric in WMNs. We focus on the fairness problem and propose FAME, an adaptive max-min fair resource allocation mechanism for WMNs. FAME provides a fair, collision-free capacity use of the WMN and automatically adjusts to the traffic demand fluctuations of the mobile clients. We develop the foundations of the mechanism and demonstrate its effi-

ciency by means of simulations. We also experimentally assess the utility of our solution when TAPs are equipped with directional antennas and distinct sending and receiving interfaces in the Magnets testbed deployed in Berlin.

In the third and last chapter of this thesis, we consider Hybrid Ad-hoc networks, i.e., infrastructured networks that are extended using multi-hop communications. We propose a secure set of protocols to encourage the most fundamental operation in these networks, namely packet forwarding. This solution is based on a charging and rewarding system. We use "MAC layering" to reduce the space overhead in the packets and a stream cipher encryption mechanism to provide "implicit authentication" of the nodes involved in the communication. We analyze the robustness of our protocols against rational and malicious attacks. We show that the use of our solution makes cooperation rational for selfish nodes. We also show that our protocols thwart rational attacks and detect malicious attacks.

Keywords

WiFi Networks, Mesh Networks, Hybrid Ad-hoc networks, Mobile wireless networks, Cooperation, Security, Pricing, Charging, Micro-payment, Network protocols, Fairness, Reputation.

Resumé

Le principe de fonctionnement de certains réseaux sans-fil nécessite la coopération des différents nœuds mobiles existants dans le réseau. Cependant, si chaque nœud est une entité égoïste et autonome, la coopération n'est pas garantie et il faut donc utiliser des mécanismes pour encourager les nœuds à coopérer. Dans cette thèse de doctorat, nous étudions la coopération dans trois différentes sortes de réseaux: Les réseaux WiFi, les réseaux Mesh et les réseaux ad-hoc hybrides. La définition de la coopération diffère d'un réseau sans-fil à l'autre, c'est pourquoi nous proposons des mécanismes d'incitation à la coopération qui sont adaptés à chacun de ces réseaux.

Dans le premier chapitre de cette thèse, nous présentons un système de réputation qui encourage le déploiement des réseaux WiFi. En effet, deux obstacles principaux continuent à ralentir le déploiement de ce genre de réseaux: d'une part, l'absence d'un system de roaming simple et efficace et d'autre part la qualité variable du service reçu par les clients mobiles. Nous proposons une solution qui (i) permet à un nœud mobile de se connecter simplement et de façon sécurisée à un hot spot Internet géré par un fournisseur d'accès étranger tout en préservant son anonymat et (ii) encourage les fournisseurs d'accès à assurer aux utilisateurs une bonne qualité de service. Le besoin pour la coopération existe ici sur deux plans distincts: D'une part, les clients mobiles doivent coopérer pour établir et maintenir le système de réputation et d'autre part, l'utilisation de ce système de réputation encourage les fournisseurs d'accès à coopérer, c'est-à-dire à fournir aux clients mobiles une bonne qualité de service. Dans ce chapitre, nous prouvons, au moyen de simulations, que notre système de réputation encourage en effet les fournisseurs d'accès à bien se comporter et nous analysons la résistance de notre solution à diverses attaques.

Dans le deuxième chapitre de la thèse, nous considérons les réseaux Mesh, un paradigme nouveau et prometteur qui utilise des relais sans-fil pour étendre la couverture des réseaux WiFi. En effet, en déployant seulement un hot spot qui est directement connecté à l'Internet et plusieurs points d'accès relais, le fournisseur d'accès Internet peut étendre la couverture qui réseau qu'il gère à moindre

coût. Nous analysons les caractéristiques des réseaux Mesh et en déduisons trois opérations fondamentales qui doivent être sécurisées : (i) la détection de relais corrompus, (ii) la définition et l'utilisation d'un protocole de routage sécurisé, et (iii) la définition et l'utilisation d'une métrique d'équité pour les réseaux Mesh.

Nous nous concentrons ensuite sur le problème d'équité et nous proposons FAME, un mécanisme adaptatif de multiplexage temporel qui assure une répartition équitable des ressources, garantie qu'il n'y ait pas de collision entre le trafic des différents clients du réseau et s'ajuste automatiquement sur les fluctuations du trafic. Nous développons le mécanisme de base de FAME et démontrons son efficacité au moyen de simulations. Nous évaluons expérimentalement l'efficacité de notre solution dans le cas où les relais sont équipés d'antennes directionnelles en utilisant Magnets, une plate-forme Mesh déployée à Berlin.

Dans le troisième et dernier chapitre de cette thèse, nous considérons les réseaux ad-hoc hybrides, c'est-à-dire des réseaux à infrastructure dont la couverture est étendue en utilisant les communications à relais. Nous proposons un ensemble de protocoles, basé sur un système de micro-paiement, qui encourage la transmission de paquets, une opération fondamentale dans ce genre de réseaux. Nous utilisons le principe de "MAC layering" pour réduire les coûts de génération et de transmission des paquets et un système d'encryptage à chiffrement de flux qui assure une "authentification implicite " des nœuds impliqués dans la communication. Nous analysons la résistance de nos protocoles à des attaques rationnelles et malveillantes. Nous prouvons, qu'en utilisant notre solution, la collaboration devient un choix rationnel pour les nœuds égoïstes. Nous prouvons également que nos protocoles résistent aux attaques rationnelles et détectent les attaques malveillantes.

Mots-clefs

Réseaux WiFi, Réseaux Mesh, Réseaux Ad-hoc hybrides, Réseaux sans-fil mobiles, Coopération, Sécurité, Micro-paiement, Protocoles réseaux, Equité, Réputation.

Acknowledgements

First of all, I would like to thank Professor Jean-Pierre Hubaux for giving me the opportunity to work at EPFL and to conduct research in his laboratory. Jean-Pierre is a great advisor who was always available and who allowed me a lot freedom in my research.

I am grateful to the members of the jury, Professor Markus Jakobsson, Professor Patrick Thiran and Professor Don Towsley, as well as to the president of the committee, Professor Emre Telatar for the time they dedicated to reading my thesis. I would like to thank Jean-Pierre Hubaux, Marcin Poturalski and Maxim Raya for reviewing all or part of this text and for their valuable feedback. I would also like to thank Holly Cogliati for improving my English and for making this thesis grammatical.

Many thanks to all the researchers with whom I worked during this thesis and in the contact of whom I have learned much. A special thanks go to Professor Levente Buttyan, Professor Anja Feldmann, Professor Markus Jakobsson, and Dr. Roger Karrer with whom I worked on papers; our conversations and interactions enriched me greatly.

I am grateful to all my past and present colleagues, thanks to whom working at LCA was such a pleasure, especially Dr. Mario Cagalj and Marcin Poturalski with whom I shared my office, and Dr. Imad Aad, Professor Srdjan Capkun, Dr. Olivier Dousse, Mathilde Durvy, Mark Felegyhazi, Dr. Jun Luo, Dr. Hossein Manshaei, Jacques Panchard, Dr. Panos Papadimistratos and Maxim Raya, with whom I had many interesting discussions.

And most of all, I want to thank my husband Khaled Grati for his love, support and patience. I am also very grateful to my parents Radhia and Mustapha, and to my brother Ghassen for their unconditional encouragement and help.

Chapter 1

Introduction

In existing wireless networks, cooperation between mobile nodes or wireless access points is, in essence, not required. However, for emerging new families of wireless networks, cooperation would be highly desirable in terms of network performance and overhead reduction. In this thesis, we consider three possible extensions of existing wireless networks and technologies, where cooperation is beneficial. The goal of each of these extensions is to propose a secure, efficient and low-cost solution that improves the network performance. Each of the proposed solutions requires the cooperation of different entities in the network. In this work, we consider the notion of cooperation in the broad sense: we define it as different nodes working together in order to perform an action that would not be possible to perform if each node behaved selfishly.

In Chapter 2, we consider existing WiFi networks and their lack of seamless roaming and quality of service (QoS) guarantees and we propose a reputation-based mechanism that encourages the Wireless Internet Service Providers (WISPs) to provide the clients with a better QoS. In our solution, the mobile clients cooperatively build a reputation system for the WISPs based on their interactions with these WISPs; the use of this reputation system encourages the WISPs to provide the mobile clients with a good QoS and offers a seamless roaming scheme.

In Chapter 3, we consider Wireless Mesh Networks (WMNs) where, by deploying only one Wired Access Point (WAP, i.e., an access point that is directly connected to the Internet), and several transit access points (TAPs), a WISP can extend its coverage and serve a large number of clients using a single broadband connection. This new kind of network allows the WISP to reduce the network deployment and maintenance costs and therefore makes it possible to lower its prices and be more competitive. Unfortunately, the medium access control (MAC) protocols used for WiFi networks are inadequate for WMNs as they lead to severe

unfairness and low bandwidth utilization. Our solution consists in a fair scheduling mechanism that assigns to each mobile client in the WMN an equal share of the network resources and optimizes the bandwidth utilization. The solution depends heavily on the cooperation among the TAPs and on the ability of the network operator to detect attacks such as intrusion or node replication.

Finally, in Chapter 4, we consider Hybrid Ad-hoc networks (also called multi-hop cellular networks). These networks can be seen as an extension of Cellular networks where multi-hop communications are used to increase the network coverage. In this new kind of network, packet forwarding is a fundamental network operation that relies entirely on the willingness of the nodes to cooperate and forward each other's traffic. In order to foster nodes' cooperation for the packet forwarding service in Hybrid Ad-hoc networks, we propose a solution based on a charging and rewarding mechanism.

In these three wireless networks, cooperation between the nodes is not guaranteed and we need to define dedicated incentive mechanisms in order to foster it. Given that cooperation depends heavily on security (see details in Subsection 1.2), each of these incentive mechanisms has to be secure against a certain number of attacks; we define in Subsection 1.3 the adversarial and threat models we consider in this thesis.

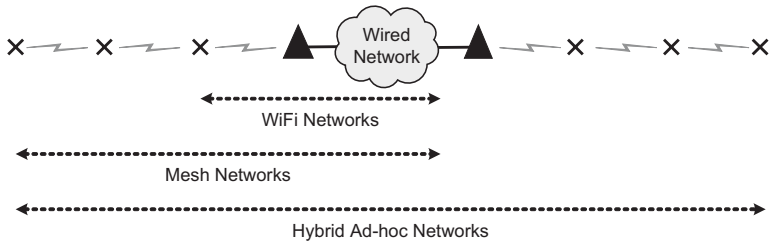


Figure 1.1: The three networks considered in this thesis.

The three kinds of networks are presented in this specific order on purpose. Indeed, given that the incentive mechanisms that we propose in this thesis have to be secure, we present the networks in an increasing order of the security challenges. As shown in Figure 1.1, in a WiFi network, a message M goes typically from the mobile client to the Internet (through the WAP) and vice versa. Therefore, M has to be secured when it is sent over the link between the mobile client and the WAP, and in the wired network. For WMNs however, M traverses a multi-hop route in order to reach the WAP or the mobile client. Thus, in addition to the protection in the wired network, M has to be secured in the multi-hop route between the mobile client and the WAP. This protection can be performed at each

hop or end-to-end, depending on the security requirements. Finally, in Hybrid Ad-hoc networks, the route from M's source to the wired network and the route from the wired network to M's destination are both multi-hop routes. The message M has thus to be secured at the wired network and in both multi-hop routes.

1.1 Cooperation in Wireless Networks

As stated earlier, we define cooperation as different nodes working together in order to perform an action that would not be possible to perform if each node behaved selfishly. Each node X wants to maximize its utility function U_X, which can be expressed as:

$$U_X = b_X - c_X$$

where b_X represents the benefit obtained by node X and c_X represents the cost of cooperation (e.g., battery consumption during packet forwarding). The benefit obtained by the nodes is expressed exclusively in terms of economy of resources, network availability, received throughput or QoS. This means that, for example, the satisfaction of harming another node is not considered as a benefit: it rather characterizes a malicious behavior (see Subsection 1.3.1).

The goal of a "regular" node is to maximize its utility function, therefore the incentive mechanism has to be designed so that cooperation becomes the best choice for these nodes. However, it is very difficult to design a perfect incentive mechanism and some nodes may find a breach in the system where they can maximize their utility function by departing from the original operation of the network. It is also possible for some nodes to pursue a different goal, e.g., instead of aiming at maximizing their utility function, they want to affect the performance of one or several other nodes in the network. We consider all the nodes that depart from the original operation of the network as attackers. We discuss the adversarial model more in details in Subsection 1.3.1.

To exemplify, let us consider peer-to-peer (P2P) networks where the importance of the cooperation between the users is obvious. In P2P networks, the payoff of an end-user depends on three main parameters:

- The probability of finding the file it searches for: A higher probability leads to a higher benefit.
- The downloading speed: The higher the speed, the higher the benefit.
- The accuracy of the information: The satisfaction level of the end-users depends on whether the data they download corresponds to the data they expect to receive.

It is clear that these three parameters depend heavily on the level of participation of the end-users. This participation can be represented by three main factors: The time they spend online, the amount of data they share, and the accuracy of these data.

We can reasonably assume that in P2P networks, the goal of a typical end-user is to download accurate data, to maximize the probability of finding this data and to maximize the downloading speed. Some end-users can have however a totally different goal: For example, if we consider that an end-user is part of the entertainment industry, its goal could be to flood the network with fake data in order to discourage the other end-users from illegally downloading songs and movies. Given that this special user aims at disrupting the operation of the P2P network, we can consider it as an attacker. Note that the definition of attacker does not take into consideration the legitimacy of the attacker's actions.

In P2P networks, the network administrator can also be part of the system and can integrate cooperation mechanisms in the P2P client in order to counter the attackers' actions, e.g., by using a reputation system, it would be possible to evaluate the behavior of the end-users or the accuracy of the data circulating in the network.

1.2 Interdependence between Cooperation and Security

In some networks (e.g., military networks), the cooperation between the nodes is guaranteed because the network is, in essence, meant to contain only cooperative nodes. However, in civilian wireless networks, each node is potentially controlled by a different entity that may be selfish, meaning that cooperation is not guaranteed anymore. If cooperation is essential for the operation of the network, incentive techniques have to be used to encourage cooperation. However, if these incentive techniques are not properly protected against cheating, they can very rapidly become useless. For example, in order to foster end-users cooperation in the P2P network Kazaa [57], the network administrators introduced in the P2P client a reputation system that evaluates the behavior of the end-user depending on the amount of data that it uploads to other users. But, the reputation of each end-user was maintained at the end-user itself; tampering with the reputation system was so easy that it became meaningless to use it [39]. Other P2P networks adapted to this reality and now some of them propose a more robust reputation system where the reputation is not stored at the node itself (e.g., eMule [79]).

These examples clearly show that cooperation and security represent two closely intricate notions. Securing the incentive mechanisms however has to be considered from a broad perspective. Indeed, ensuring the integrity of the messages used to

establish and maintain a reputation system is neither sufficient nor efficient in a wireless network where simple and powerful attacks such as Sybil attack, node duplication or intrusion are possible.

1.3 Security in Wireless Networks

1.3.1 Adversarial Model

We define an adversary as a node that intentionally deviates from the original operation of the network. The adversary's actions aim at disrupting one or several of the following security objectives:

- Data integrity: Ensuring data has not been altered.
- Node and data authentication: Verifying the identity of a node or the origin of the data.
- Data confidentiality: Keeping the data secret from all except those who are authorized to see it.
- Non-repudiation: Preventing the denial of previous commitments or actions.
- Node anonymity: Concealing the identity of an entity involved in some process.
- Availability: Maintaining the system operational.

An attacker $\mathcal{A}$ is *rational* if its goal is to increase its utility function and if, by cheating, it can do so, e.g., by receiving extra payments, more service or by saving resources. Otherwise, $\mathcal{A}$ is *malicious*. Note that if each node represents an independent and selfish entity and if some aspects of the solution make it possible to increase a node's utility function by departing from the original set of protocols, then every node in the network is potentially an attacker.

We assume that several attackers can collude to perform more sophisticated attacks. We also assume that an attacker is occasionally able to compromise nodes by retrieving their secret information.

For the three incentive mechanisms we present in this thesis, we will consider exclusively attacks performed against the different phases of our protocols, meaning that we do not consider other arbitrary attacks such as Denial of Service (DoS) attacks based on jamming. However, in some cases, DoS attacks can be rational. For example, as we will see in Chapter 3, a Wireless Internet Service provider (WISP) can jam access points managed by its competitors to disrupt its service and

to get its clients. Therefore, we design our solutions with DoS in mind, meaning that we define sets of protocols where the effect of these attacks is minimal and where we can rapidly detect the attack. We also make sure that we do not expose protocol participants to unnecessary risks by relying on heavyweight operations.

1.3.2 Threat Model

We distinguish between *passive* and *active* attacks. In passive attacks, the attacker analyzes the data without altering them (we also refer to these kinds of attacks as *eavesdropping* attacks), whereas in active attacks, the attacker modifies, deletes or injects data in the network.

There is a large variety of active attacks; some are *specific* to the incentive mechanism and the others are *general* attacks that are independent of it. We identify the following general attacks:

- *Packet Dropping Attack*: $\mathcal{A}$ drops a packet it is asked to forward.
- *Filtering Attack*: $\mathcal{A}$ modifies a packet it is asked to forward.
- *Replay Attack*: $\mathcal{A}$ replays a valid packet out of its legitimate context.
- *Emulation Attack*: $\mathcal{A}$ uses the secret key of a node
- *DoS attacks*: $\mathcal{A}$ prevents two or more nodes from communicating, e.g., by jamming the wireless channel.
- *Sybil Attack*: $\mathcal{A}$ has different identities
- *Intrusion Attack*: $\mathcal{A}$ is an unauthorized node but it manages to be accepted in the network as a valid node.
- *Node Duplication Attack*: $\mathcal{A}$ creates one or several instances of a node. This attack is also referred to as cloning attack.

We will specify the list of specific attacks in each of the chapters.

Chapter 2

Reputation System for WiFi Networks

2.1 Introduction

The rapid growth of WiFi networks over the past years is due primarily to the fact that they solve several of the intrinsic drawbacks of cellular data services such as GSM/GPRS/UMTS. These drawbacks are mainly the relatively low offered bitrates and the slow deployment of new features due to several factors such as the large size and the oligopolistic behavior of the operators, their willingness to provide homogeneous service, and the huge upfront investment. Therefore, the deployment of wireless networks such as WiFi in unlicensed frequencies makes it possible to envision a substantial *paradigm shift*, with very significant benefits: much higher bandwidth, deployment based possibly on local initiative, higher competition, and much shorter time-to-market for new features. This may, in turn, pave the way for new types of services.

In recent years, wireless Internet service providers (WISPs) have installed thousands of WiFi hot spots notably in cafes, hotels and airports. However, two major problems still need to be solved. The first problem is the provision of a seamless roaming[1] scheme that would encourage small operators to enter into the market. This is a fundamental issue for the future of mobile communications. Indeed, without an appropriate scheme, only large stakeholders would be able to operate their network in a profitable way, and would impose a market organization very similar to the one observed today for cellular networks; one of the greatest opportunities to

[1]Note that by roaming we designate the operation of obtaining service from different operators, and not the handoff between access points (whether managed by the same provider or by two different providers). The handoff problem is out of the scope of this work.

fuel innovation in wireless communications would be missed. The second problem is the lack of a good quality of service guarantee for the users.

This chapter provides a response to these challenges. By appropriately unbundling the major functions of the network, our solution institutes a virtuous cycle of deployment and usage: Each WISP will be encouraged to deploy its network and will be confident that mobile users registered with other WISPs will pay for the service it provides them; likewise, users will be assured that the WISPs are under the scrutiny of all the other users (including the roaming ones), and that they will be informed about their degree of satisfaction.

As we will see, the solution is relatively simple, provided that the roles of the different entities are clearly defined. We describe these entities in detail, along with the security protocols and the charging mechanism. In order to facilitate user acceptance, the proposed solution minimizes user involvement: once the mobile device has been initialized, it can make all decisions autonomously.

One of the major goals of this work is to build up trust between mobile users and WISPs. For this reason, we provide a detailed threat analysis and we show that the proposed protocols can thwart rational attacks and detect malicious attacks (we define these terms in Subsection 2.2.2).

Outline This chapter is organized in the following way: In Section 2.2 we present the system and trust models and in Section 2.3, we give an overview of the proposed solution and describe the details of the protocols. We evaluate the reputation system by means of simulations in Section 2.4 and we study the security of the protocols in Section 2.5. We evaluate the overhead in Section 2.6. Finally, we present the state of the art in Section 2.7 and we conclude in Section 2.8.

2.2 System and Trust Models

2.2.1 System Model

In this chapter, we consider a mobile node (*MN*) that wants to connect to the Internet via a neighboring hot spot (i.e., a hot spot that is within its power range); we assume the hot spot to be managed by a WISP that we denote by S (see Figure 2.1). *MN* is affiliated with its home WISP[2] H with whom it has an account and shares a symmetric key k_{HM}. We assume that all the messages exchanged between *MN* and H go through S, however, we ensure *MN*'s anonymity with respect to a foreign WISP S (note that it is possible to have $S = H$).

[2]The solution works even if H does not operate hot spots itself.

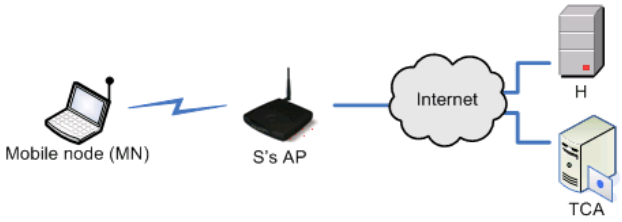


Figure 2.1: System model.

In our model, all WISPs are registered with the trusted central authority (*TCA*) that creates for each of them a public/private key pair and a certificate of their public key and of their identity. We assume that *TCA*'s public key is known by all other entities. In a "grassroots" vision, *TCA* would be a federation of WISPs, who join forces to centralize a few strategic functions. In a more conventional vision, *TCA* can be under the control of a world-wide organization such as a quality control company, a certification company, or a global telecommunications operator. *TCA* servers can be distributed to avoid bottlenecks.

In this work, we present a reputation based mechanism that, on the one hand, allows *MN* to evaluate the behavior of the WISPs and, on the other hand, encourages the WISPs to provide the users with good QoS. In our model, each WISP has what we call a *reputation record* that represents an evaluation of its behavior and that is generated and signed by *TCA*. The choice of the initial reputation record of a WISP is discussed in Section 2.4.

2.2.2 Trust and Adversarial Model

We consider an attacker $\mathcal{A}$ that wants to perform an attack against our protocols (see Section 2.5 for the list of attacks). $\mathcal{A}$ can be a mobile node or a WISP. We assume that (i) the *TCA* never cheats and is trusted by the other parties for all the actions it performs; (ii) the WISPs (here S and H) are rational and therefore they cheat (i.e., perform one of the attacks presented in Section 2.5) only if it is to their advantage (e.g., in terms of money); and (iii) *MN* may be malicious and therefore it can cheat even if there is no gain from cheating (this implicitly assumes that *MN* can also perform rational attacks). We also assume that *MN* trusts H to manage its account and that several attackers can collude and share information (possibly their secret keys) to perform more sophisticated attacks.

In this chapter, we want to study the effect of rational and malicious attacks on our set of protocols. Our goal is to make sure that our solution thwarts rational attacks, detects malicious attacks and, if possible, identifies the attacker.

Confidentiality of data is not an issue in our case, so we do not consider passive attacks where the attacker eavesdrops the data exchanges between two parties. Note that this is an orthogonal issue that is easily addressed using standard security techniques.

2.3 Details of the Protocols

2.3.1 Rationale of the Solution

Our solution consists of four phases: *Session Setup*, *Service Provision and Payment*, *Session Closing* and *Reputation Update*.

Session Setup: When *MN* wants to connect to the Internet, it contacts all the neighboring WISPs[3] and selects the WISP S that presents the best offer. The decision making is based, among other criteria, on the reputation records of the WISPs (see Subsection 2.3.3). Then, *MN* and S establish a secure session by setting up a symmetric key k_{MS}.

Service Provision and Payment: This secure session is divided into parts. During the i-th part, *MN* sends a payment proof for the i-th part of the service and S provides that part of the service. In order to make sure that the mobile nodes pay for the service they receive, we use a credit-based micro-payment scheme: the PayWord scheme [76] (see Subsection 2.3.2).

Session Closing: At the end of the connection, the session is closed and *MN* reports on the QoS it received to *TCA*.

Reputation Update: *TCA* collects the feedback about the different WISPs, updates periodically the reputation records according to the collected information, and provides the WISPs with their new reputation records.

2.3.2 Basic Mechanisms

Micro-payment scheme

As already mentioned in Section 2.2, the payment scheme we use in this work is the PayWord scheme [76]:

During the session setup, *MN* generates a long fresh chain of paywords w_0, w_1, ..., w_n by choosing w_n at random and by computing $w_i = h(w_{i+1})$ for $i = n-1, n-2, \ldots, 0$, where h is a one-way hash function and n is the maximum number of payments that *MN* can send to S during the session. Then, *MN* reveals the root w_0 of the payword chain (which is not considered as a payword itself) to S, H and *TCA*.

[3] Note that we refer to the access points using the identities of the WISPs that are managing them.

During the secure session, *MN* sends (w_i , i) to S as a payment proof for the i-th part of the service. S can easily verify w_i using w_{i-1} that is known from the previous micro-payment or from w_0 if $i = 1$.

At the end of the session, S sends the last payment (w_ℓ , ℓ) it received to H. H verifies the validity of w_ℓ, pays S the amount corresponding to ℓ paywords and charges *MN* for that amount by updating its billing account.

We use this micropayment scheme because it allows for an offline verification of the payment proofs and because of its low computational and storage costs for the mobile nodes.

Authentication of *MN* by *H*

As stated in Section 2.2, all communication between *MN* and H goes through S. Therefore, in order to preserve the anonymity of *MN* with respect to S, we use the following authentication mechanism, which is commonly used in the industry (e.g., SecurID [46]): When *MN* gets affiliated with H, the two parties share a random seed s that represents the input to a pseudorandom generator. The output is a random number *tag* that is 30 to 50 bits long. H keeps a small window (e.g., 50 entries) of upcoming tags for each mobile node and maintains the pairs (*tag; node's identity*) in a sorted database. Upon receipt of a given *tag*, H searches its database, retrieves the pair (*tag;identity*) and identifies *MN*. In case of collision (i.e., more than one pair contains the random number *tag*), H asks *MN* to send the next tag value.

2.3.3 Details of the Protocols

Session Setup Phase

This phase consists of three steps (see Figure 2.2): *Selection of the WISP*, *Authentication of* MN and *Secure session establishment*.

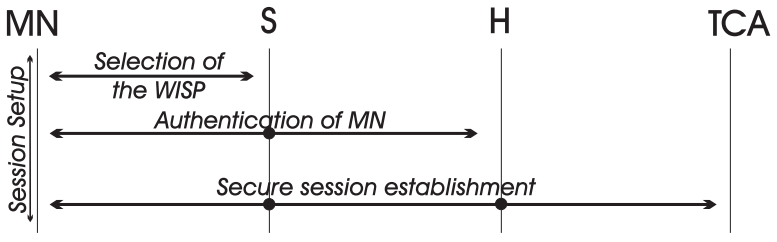


Figure 2.2: Session setup

Selection of the WISP: When *MN* wants to obtain Internet access, it scans the spectrum, contacts the neighboring WISPs and asks for an offer by broadcasting the following request message:

$$\mathit{OfferReq} = [\mathit{ReqID}, N_M] \tag{2.1}$$

where ReqID is the request identifier and N_M is a nonce generated by MN. Each WISP W willing (and able) to provide service at that time responds by a signed offer Offer_W:

$$W \rightarrow \mathit{MN} : \mathit{Offer}_W, S_{pk_W}(\mathit{Offer}_W, \mathit{OfferReq}) \tag{2.2}$$

where

$$\mathit{Offer}_W = [W, RR_W, AQ_W, Pr_W, \mathit{Cert}(W), N_W]$$

RR_W is the most recent *reputation record* of W (signed by *TCA*), AQ_W is the QoS it advertises[4], Pr_W is the price it is requesting for each part of the service (see Subsection 2.3.3), pk_W is its private key and *Cert*(W) is the certificate of its public key PK_W.

For each offer Offer_W, MN verifies the freshness of n_W and computes a value[5] $\mathit{Decision}_W = RR_W^{\alpha} \cdot AQ_W^{\beta} \cdot P_W^{-\gamma}$, where the exponents α, β and γ are parameters that depend on the application *MN* is running[6]. Then, *MN* determines $\mathit{Decision}_S = \max_W\{\mathit{Decision}_W\}$, selects the WISP S and verifies its certificate and the signature of its offer. If the verification is incorrect, *MN* checks the second best offer and so on. We denote the selected WISP by S.

Authentication of *MN*: Before starting the session, S has to make sure that *MN* is a valid mobile node that is registered with a valid home WISP. As we want to preserve the anonymity of *MN*, the verification of *MN*'s identity involves H and uses the authentication mechanism described in Subsection 2.3.2. We have thus the following messages exchanged:

$$\begin{aligned} \mathit{MN} \rightarrow S \quad &: \quad \mathcal{M} = [H, tag, N_M \\ &\qquad E_{k_{HM}}(\mathit{MN}, S, tag, N_M)] \qquad (2.3) \\ S \rightarrow H \quad &: \quad S, N_S, \mathcal{M}, MAC_{k_{HS}}(S, \mathcal{M}) \qquad (2.4) \end{aligned}$$

[4]The estimation of the QoS offered by W is discussed in Section 2.4.4.

[5]The decision function given here is an example; it can be any function $f(RR_W, AQ_W, P_W)$.

[6]We can have for instance (α,β,γ)= (2,1,3) for chat applications to put the emphasize on low price offers and (α,β,γ)= (2,2,1) for file transfer applications to put the emphasis on QoS. The decision function being exponential amplifies the difference between these two cases.

$$H \rightarrow S \quad : \quad TID, E_{k_{HM}}(TID, N_M, k_{MS}), \\ E_{k_{HS}}(TID, N_S, k_{MS}) \tag{2.5}$$

$$S \rightarrow MN \quad : \quad TID, E_{k_{HM}}(TID, N_M, k_{MS}) \tag{2.6}$$

(2.3) *MN* sends to S a message $\mathcal{M}$ containing, in clear, the identity of H, its current *tag* and a freshly generated nonce N_M. $\mathcal{M}$ also contains, encrypted using the symmetric key[7] k_{HM}, the identities of *MN* and S, the tag *tag* and the nonce N_M.

(2.4) S sends to H its identity, a freshly generated nonce N_S, the message $\mathcal{M}$ and a MAC computed on both items using the key k_{HS}.

(2.5) H searches its sorted database, identifies *MN* using the *tag* sent in clear (as explained in Subsection 2.3.2), looks up the symmetric key it shares with *MN* and uses it to decrypt the rest of the message. Then, H re-checks the identity of *MN* (the identity corresponding to the tag should also correspond to the identity *MN* encrypted in the message) and verifies that the WISP with which *MN* intends to interact is indeed the WISP that sent the message.

If the message is not correct, H informs S that *MN* is not affiliated with it by sending a negative acknowledgement. If, on the contrary, the message verifies correctly, H generates a symmetric key k_{MS} that *MN* and S will use later as a session key (i.e., all the messages exchanged between *MN* and S during the session are secured using k_{MS}). Then, H constructs a message containing (i) in clear, a fresh temporary identifier TID for *MN* (TID will be used during service provision), (ii) TID, N_M, and k_{MS} encrypted using the symmetric key k_{HM}, and (iii) TID, N_S, and k_{MS} encrypted using the symmetric key k_{HS}, and sends this message to S. H maintains a table containing the correspondence between the temporary identifiers and the identities of the nodes; given TID, H can identify the correspondent *MN*.

(2.6) S decrypts $E_{k_{HS}}(TID, N_M, k_{MS})$, verifies that the temporary identifier in the decrypted part corresponds to the one sent in clear, and compares the nonce in the decrypted part with the one generated by *MN*. If these verifications are correct, S removes $E_{k_{HS}}(TID, N_M, k_{MS})$ from the message and forwards the rest to *MN*.

MN decrypts $E_{k_{HM}}(TID, N_H, k_{MS})$ and verifies the temporary identifier and the nonce as S did. If everything is correct, *MN* maintains *TID* in memory.

Note that if $S = H$, *MN* sends message (2.3) to H and H responds with message (2.6).

Secure Session Establishment: *MN* generates a long hash chain of $n + 1$ elements, computed from a randomly chosen seed w_n as described in Subsection 2.3.2.

[7] H and S can use their public keys to establish a temporary symmetric key k_{HS}. We assume that this key is generated prior to the execution of our set of protocols.

Then *MN* generates a contract

$$C = [CID, w_0, RR_S, AQ_S, Pr_S]$$

where $CID = [TID, S, H]$ is the contract identifier and w_0 is the root of the hash chain.

Then *MN* and S inform H about the contract:

$$MN \rightarrow S \quad : \quad C, MAC_{k_{MS}}(C), MAC_{k_{HM}}(C) \tag{2.7}$$
$$S \rightarrow H \quad : \quad C, MAC_{k_{HM}}(C), MAC_{k_{HS}}(C) \tag{2.8}$$

(2.7) *MN* sends the contract C to S, together with two MACs computed on C using the symmetric keys k_{MS} and k_{HM}, respectively.

(2.8) S verifies C and $MAC_{k_{MS}}(C)$ and if they are correct, it computes a MAC on C using the symmetric key k_{HS} it shares with H. Then, S sends to H the contract C and the MACs computed with k_{HM} and k_{HS}. H verifies the MACs and, if they are correct, it stores the contract C.

MN and S also inform TCA about the contract:

$$MN \rightarrow S \quad : \quad E_{PK_{TCA}}(C, k_{MT}), \; MAC_{k_{MS}}(E_{PK_{TCA}}(C, k_{MT})) \tag{2.9}$$
$$S \rightarrow TCA \quad : \quad C, E_{PK_{TCA}}(C, k_{MT}) \tag{2.10}$$
$$TCA \rightarrow S \quad : \quad S_{pk_{TCA}}(C), MAC_{k_{MT}}(C) \tag{2.11}$$
$$S \rightarrow MN \quad : \quad MAC_{k_{MT}}(C) \tag{2.12}$$

(2.9) *MN* generates a fresh symmetric key k_{MT} that *MN* will use later to encrypt data for *TCA* (see Subsection 2.3.3). Then, *MN* encrypts[8] C and k_{MS} using the public key of *TCA*, computes a MAC on the data using k_{MS} and sends the encrypted data and the MAC to S.

(2.10) S verifies the MAC and sends C and the encrypted data to *TCA*.

(2.11) *TCA* decrypts the data and compares the contract C received in the encrypted data with the contract received in clear from S. If they are identical, *TCA* signs the contract C using its private key pk_{TCA}, computes a MAC on it using the symmetric key k_{MT} that is shares with *MN*, and sends the signature and the MAC back to S. *TCA* also maintains C and k_{MT} in its local database.

(2.12) S verifies *TCA*'s signature and forwards the MAC to *MN*.

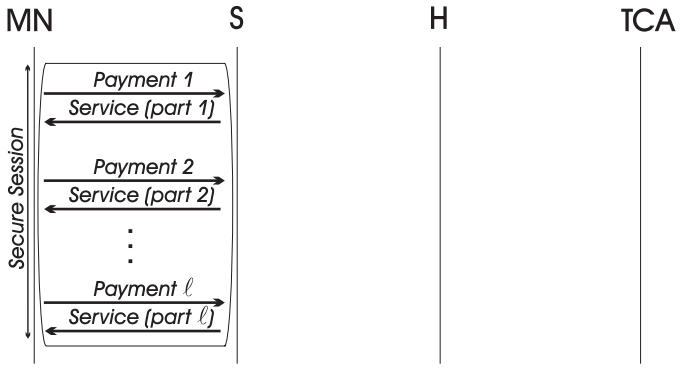


Figure 2.3: Service provision and payment

Service Provision and Payment

The session is subdivided into parts, depending on the duration or on the amount of data exchanged between *MN* and S. During the i-th part:

$$MN \rightarrow S \quad : \quad TID, w_i, MAC_{k_{MS}}(TID, w_i) \tag{2.13}$$

$$S \rightarrow MN \quad : \quad i^{th} \text{ part of the service} \tag{2.14}$$

(2.13) *MN* sends to S its temporary identity TID, the i-th PayWord w_i and a MAC computed on both items using the key k_{MS}.

(2.14) S verifies the validity of w_i by checking that $h(w_i) = w_{i-1}$, where h is the one-way hash function used by *MN* to generate the chain. If it is correct, S provides *MN* with the i-th part of the service. Note that the data packets corresponding to the i-th service are cryptographically protected using the key k_{MS} (e.g., the key is used to encrypt the packets if privacy is required and to compute a MAC if authentication is required).

Session Closing and Reputation Update

At the end of the session, S sends to H a payment request *PR* that contains, encrypted using k_{HS}, the contract identifier *CID*, the last hash value w_ℓ it received from *MN* and the number ℓ of provided service parts. *PR* also contains, in *CID*, the identity of S so that H is able to retrieve the symmetric key k_{HS}.

$$S \rightarrow H : PR = [CID, w_\ell, \ell, MAC_{k_{HS}}(CID, w_\ell, \ell)] \tag{2.15}$$

[8] In order to prevent the key retrieval by an attacker, *MN* can use a probabilistic encryption algorithm, e.g. RSA-OAEP [9], RSA-PSS [10] or ElGamal [33].

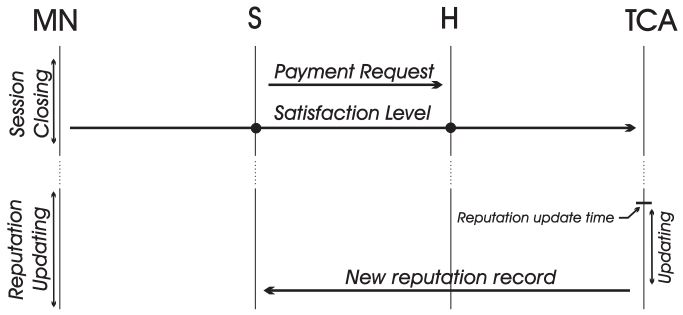


Figure 2.4: Session closing and the reputation update

Upon receipt of *PR*, H verifies the validity of w_ℓ as explained in Subsection 2.3.2, retrieves the price Pr_S from the contract, rewards S for the ℓ parts of the service, and charges *MN*. H is also remunerated (see details in Subsection 2.3.4).

At the end of the session, *MN* generates a *satisfaction level* message Sl as follows:

$$Sl = [E_{k_{MT}}(CID, QoSEval_{S,CID}, w_\ell, \ell)] \tag{2.16}$$

$QoSEval_{S,CID}$ is *MN*'s estimate of the compliance of the obtained QoS with the announced one and k_{MT} is the key *MN* shares with *TCA*.

Then, *MN* sends its *satisfaction level* to *TCA*:

$$MN \rightarrow S \quad : \quad TID, Sl, MAC_{k_{MS}}(TID, Sl) \tag{2.17}$$

$$S \rightarrow TCA \quad : \quad S, CID, w_\ell, \ell, Sl, S_{PK_S}(S, CID, w_\ell, \ell, Sl) \tag{2.18}$$

(2.17) *MN* sends to S its temporary identifier *TID*, *Sl* data and a MAC computed on both items.

(2.18) S verifies the MAC. If it is correct, S generates a message containing *CID*, w_ℓ, ℓ and *Sl*, signs it and sends the message and the signature to *TCA*.

TCA verifies the signature and retrieves the key it shares with *MN* (using *CID*). Then *TCA* decrypts *Sl*, compares the *CID*, w_ℓ, ℓ in the encrypted data to those received in clear from S and if they are identical, *TCA* considers *QoSEval* as a valid feedback. Then *TCA* informs H that it correctly received the feedback:

$$TCA \rightarrow H : Ack, S, CID, S_{PK_{TCA}}(Ack, S, CID) \tag{2.19}$$

(2.19) H verifies the signature and retrieves the identity of *MN* (using *CID*). Then, H remunerates *MN* a small amount of money ε, which is meant to encourage the mobile nodes sending the reports.

TCA collects the information about the satisfaction levels for a given period and then, at the *reputation update time*, it updates the reputation record of each WISP, signs them and informs the WISPs about their new records. The new reputation record depends on the old one and on the collected information. An example is given in Subsection 2.4.

TCA considers the absence of feedback as negative feedback. Indeed, *TCA* knows that a session has been established between *MN* and S and that H is the home WISP of *MN* (see Subsection 2.3.3). *TCA* is thus waiting for the report from *MN* about its interaction with S, and not receiving it within a "reasonable" time is considered as bad feedback.

2.3.4 Charging and Rewarding Model

In this subsection, we provide additional details regarding the charging and rewarding model:

- If, at the end of the session, *MN* moves away from S (and therefore cannot send the feedback via S), it is still possible for *MN* to report on its satisfaction level to *TCA* via another WISP W: W includes its identity in message (2.18) and signs the message using its own private key. *TCA* then verifies the signature and informs H in message (2.19) about the identity of W. Then H gives both *MN* and W a reward $\varepsilon/2$.

- At the end of the session, S sends to H the last payment proof (w_ℓ, ℓ) it received from *MN*. H verifies the validity of the payword w_ℓ, charges *MN* the amount $X = Pr_S * \ell$ corresponding to the ℓ parts of the service and rewards S, using a well-established e-payment technique, the amount[9] $X - \varepsilon$. If *TCA* receives no report from *MN*, ε is handled according to some policy (e.g. it can be distributed to charity).

- The home network H is also remunerated. This can be done e.g., if *MN* pays a flat monthly subscription or if *MN* pays a specific amount per session.

2.4 Analysis of the Incentive Mechanism

Our solution motivates the different players to participate in the reputation mechanisms. Indeed (i) S is motivated to provide *MN* with the QoS it promised because otherwise the feedback of *MN* will be negative (see the analysis of the

[9] As already mentioned in Subsection 2.3.3, ε is the reward *MN* receives if it reports on its satisfaction level to *TCA*.

Publicity attack in Subsection 2.5.1), (ii) *MN* is motivated to report on its interaction with S because it receives a refund ε and (iii) S is motivated to forward the report (see the analysis of the report dropping attack in Subsection 2.5.1).

However, we want also to study the effect of the reputation mechanism on the behavior of the WISPs, i.e., the QoS they effectively offer to the mobile users. We therefore implemented our set of protocols using the ns-2 simulator [45].

2.4.1 Simulation Environment

We consider a static[10] network of 5 WISPs, numbered from 1 to 5, and 50 MNs. Each WISP is a home WISP for 10 MNs. Each WISP W is characterized by a triplet (AQ_W,RQ_W,P_W) where AQ_W is the QoS advertised by W, RQ_W is the real QoS provided by W and P_W is the price W is asking for. We consider that a WISP W is *honest* if it advertises the real QoS it is offering (i.e., $RQ_W = AQ_W$), *misbehaving* if it advertises a QoS that is higher that the real QoS it is offering (i.e., $RQ_W < AQ_W$) and *modest* if it advertises a QoS that is lower than the real QoS it is offering (i.e., $RQ_W > AQ_W$).

We initialize the reputation of the WISPs to $maxRR = 100$. At the end of each session, MN sends to TCA its *satisfaction level*

$$Sl = [E_{k_{MT}}(CID, QoSEval_{W,CID}, w_\ell, \ell)]$$

where

$$QoSEval_{W,CID} = \frac{RQ_W}{AQ_W}$$

Each simulation lasts for 50000 seconds and the reputation updates are made every 2000 seconds. The new reputation $RR_W(t+1)$ of S is computed as follows:

$$RR_W(t+1) = \lambda \cdot RR_W(t) + (1-\lambda) \cdot \frac{feedback_W}{nbS_W}$$

where $RR_S(t)$ is the current reputation of W, nbS_W is the number of sessions established by W (and already closed) during the last 2000 seconds and $feedback_W$ is the sum of all $QoSEval_{W,CID}$ received over all these sessions (the absence of feedback is considered as $QoSEval_{W,CID} = 0$). λ represents the "weight of the past" and is set to $1/2$ in our simulations.

Note that if S advertises a QoS that is lower than the real QoS it offers (i.e., $AQ_W < RQ_W$), we will have $QoSEval_W > maxRR$, which may lead to a new reputation that is also higher than $maxRR$. If it is the case, *TCA* keeps

[10] All *MN*s are within the power range of all WISPs, it is therefore useless to consider mobility in this case.

$RR_W(t+1)$ as it is in its database but sends to S a new reputation record equal to $maxRR$.

2.4.2 Studied Scenarios

We consider a network of 5 WISPs and 50 MNs. The WISPs are numbered from 1 to 5 and for each WISP, we define the advertised QoS, the real QoS and the price it asks for each part of the service. We initialize the reputation of the WISPs to $maxRep = 100$. *MN*s and WISPs are static[11] and each WISP is a home WISP for 10 MNs. Each simulation lasts for 50000 seconds and the reputation updates are made every 2000 seconds.

We conducted three sets of simulations to study three aspects of our solution:

Set 1: We want to study the reaction of the network if all the WISPs are honest but offer different QoSs: WISPs 1, 2, 3, 4 and 5 advertise and offer QoS = 60, 70, 80, 90 and 99, respectively[12]. We consider the two following scenarios:

- **Scenario 1.1:** All the WISPs ask for the same price. At the beginning of a simulation, we assign to each *MN*, with equal probability, one of the two following applications: chat or file transfer (see Subsection 2.3.3).
- **Scenario 1.2:** The WISPs ask for prices that are proportional to their QoSs ($P_W \sim RQ_W$). We expect the choice of the application to have an effect on the results, so we run 2 sets of simulations; one for each kind of application (i.e., all the nodes run that application).

Set 2: We want to study the reaction of the network to the co-existence of honest, misbehaving and modest WISPs in the network: WISPs 1, 2, 3, 4 and 5 advertise AQ = 60, 70, 80, 90 and 99, respectively; but all of them offer $RQ = 80$. We consider the two following scenarios:

- **Scenario 2.1:** All the WISPs ask for the same price. At the beginning of a simulation, we assign to each *MN*, with equal probability, one of the following applications: chat or file transfer.
- **Scenario 2.2:** The WISPs ask for prices that are proportional to their QoSs ($P_W \sim RQ_W$). We expect the choice of the application to have an effect on the results, so we run 2 sets of simulations; one for each kind of application (i.e., all the nodes run that application).

[11] All *MN*s are within the power range of all WISPs, it is therefore useless to consider mobility in this case.

[12] We do not consider the case where $AQ = 100$ because such a perfect case is probably not possible in real life conditions.

Set 3: We assume that all the WISPs are honest, offer the same QoS and ask for the same price. At the beginning of a simulation, we assign to each *MN*, with equal probability, one of the following applications: chat or file transfer.We want to study the effect of the initial reputation of a WISP that opens its service. We assume that the newcomer is WISP 1 and we consider the three following scenarios:

Scenario 3.1: The initial reputation of WISP 1 equals the one of the other WISPs ($Rep_1 = maxRep = 100$ because the WISPs are honest).

Scenario 3.2: The initial reputation of WISP 1 is lower than the one of the other WISPs ($Rep_1 = 50$).

Scenario 3.3: The initial reputation of WISP 1 is lower than the one of the other WISPs ($Rep_1 = 50$) but WISP 1 asks for a lower price.

For all these scenarios, we assume that the values of AQ and RQ remain constant and are independent from the number of *MN*s that are simultaneously connected to the WISPs[13].

2.4.3 Simulation Results

We run 10 simulations for each of the scenarios listed in Subsection 2.4.1. The results are the following:

Set 1: The results of Scenario 1.1 show that if all the WISPs ask for the same price, most of the users select the WISP that offers the best QoS (WISP 5 in Figure 2.5). The other WISPs (mainly WISP 4) can occasionally have some clients because the randomness introduced for the service provision at the WISPs (see Subsection 2.4.2) may lead to a slight decrease in WISP 5's reputation.

The results of Scenario 1.2 show that if all the WISPs offer different QoSs and ask for different prices, the choice of the users depends on the application they are running; e.g., if the nodes run a chat application (see Figure 2.6), the majority of the nodes choose the WISP 2 whereas if the nodes run a file transfer application (see Figure 2.7), the majority of the nodes choose the WISP 5 that offers the best QoS.

Note that in Scenario 1.2, nodes running the chat application do not choose WISP 1 even if it offers a lower price than WISP 2. By analyzing the data, we realized that this is because the reputation of WISP 2 is significantly higher than the one of WISP 1, which is caused by the randomness introduced, for the service provision, at the WISPs (see Subsection 2.4.2).

[13]The case where these values vary is studied in Section 2.4.4.

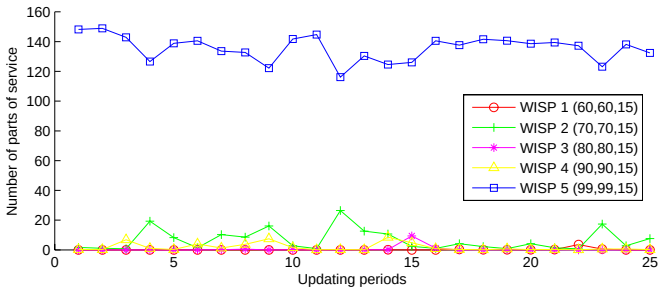


Figure 2.5: Scenario 1.1: All the WISPs are honest and ask for the same price. Therefore, WISP 5, which offers the highest QoS, eventually gets most of the users.

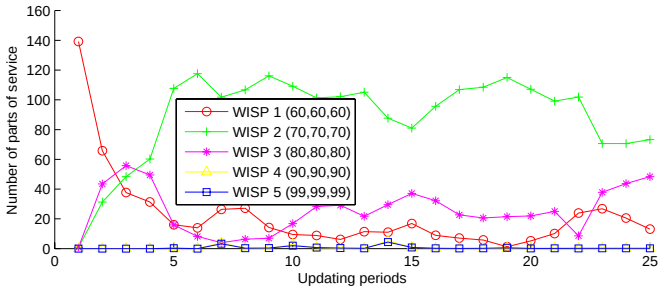


Figure 2.6: Scenario 1.2: All the nodes run a chat application. They choose WISP 2 which asks for a low price and at the same time has a good reputation.

These results clearly prove that:

- the WISPs are encouraged to provide a good QoS and
- honest WISPs offering different QoSs can co-exist in the same network.

Set 2: The results of Scenario 2.1 show that if all the WISPs ask for the same price, most of the users select the WISP with the advertised QoS that corresponds best to the real QoS it offers (WISP 3 in Figure 2.8). Due to their good reputation, modest WISPs (here WISPs 1 and 2) perform better than misbehaving WISPs (here WISPs 4 and 5) but are still selected much less often than the honest WISP. Indeed,

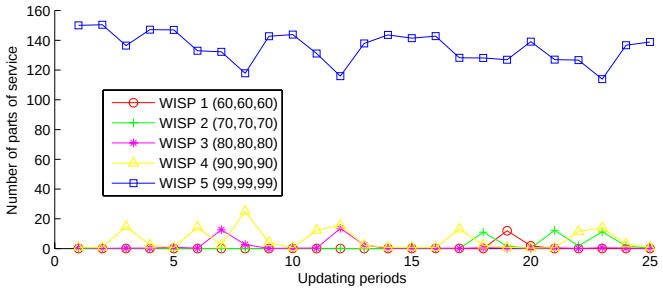


Figure 2.7: Scenario 1.2: All the nodes run a file transfer application. They choose WISP 5 because it offers the best QoS.

among the WISPs that have good reputations (WISPs 1, 2 and 3), WISP 3 is the one offering the best QoS and thus is selected more often. Therefore, the best strategy for the WISPs is to be honest.

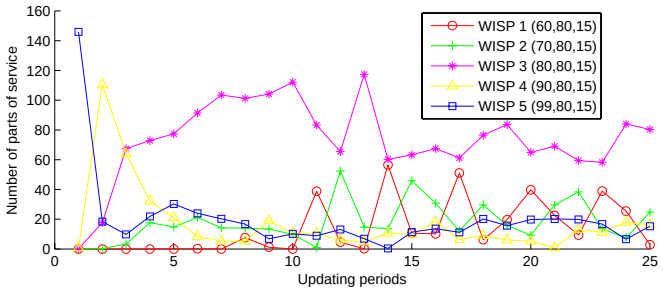


Figure 2.8: Scenario 2.1: All the WISPs ask for the same price. The only honest WISP, here WISP 3, eventually gets most of the users.

Note that the mobile nodes have no direct indication on the real QoS of the WISPs. They are however able to correctly evaluate the behavior of the WISPs because the correspondence between the advertised QoS and the real QoS is taken into consideration in the updating of the reputations.

The results for Scenario 2.2 show that almost all the nodes that run the chat application (see Figure 2.9) choose WISP 1, which asks for the lowest price and at

the same time has a very good reputation. The majority of the nodes running a file transfer application (see Figure 2.10) choose WISP 3 because it offers the best real QoS.

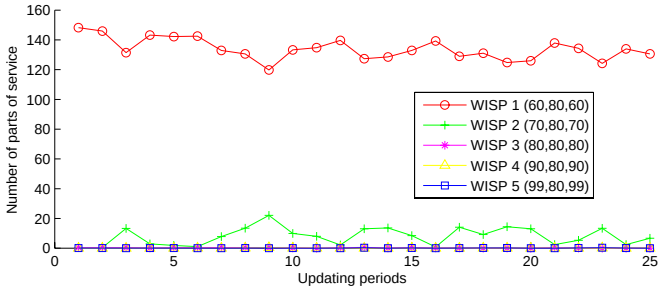


Figure 2.9: Scenario 2.2: All the nodes run a chat application. They choose WISP 1 because it asks for the lowest price and at the same time has a good reputation.

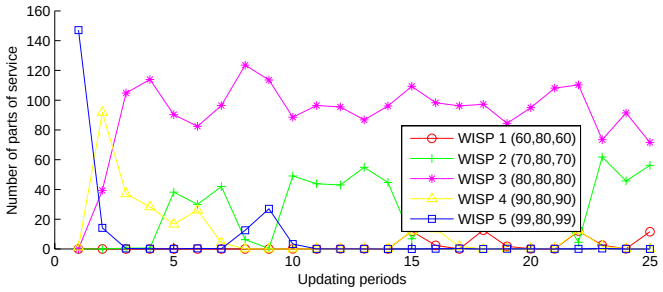


Figure 2.10: Scenario 2.2: All the nodes run a file transfer application. They choose WISP 3 because it offers the best real QoS.

These results clearly prove that the WISPs are discouraged from misbehaving (i.e., to advertise a QoS that is higher than the real QoS they can offer) and from being modest (i.e., advertising a QoS that is lower than the real QoS they can offer).

Set 3: In Scenarios 3.1 and 3.2, all the WISPs offer the same QoS and ask for the same price.

The results of Scenario 3.1 show that if WISP 1 has, when it opens its service, the same reputation as the other WISPs, it has more or less the same probability to get clients as others do (see Figure 2.11).

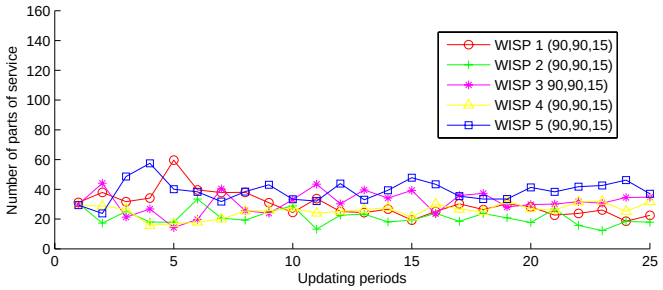


Figure 2.11: Scenario 3.1: WISP 1 has, when it opens its service, the same reputation as the other WISPs ($Rep = 100$); it has more or less the same probability to get clients as others do.

The results of Scenario 3.2 show if WISP 1 has, when it opens its service, a reputation that is lower than the reputation of all other WISPs, it has no chance to get clients. (see Figure 2.12).

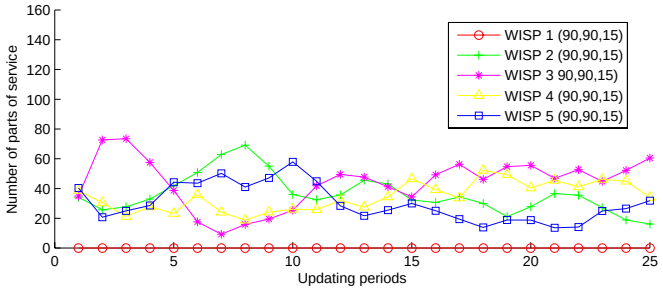


Figure 2.12: Scenario 3.2: WISP 1 has, when it opens its service, a lower reputation than for the other WISPs; it has no chance to get clients.

In Scenario 3.3, all the WISPs offer the same QoS and all of them, except WISP 1, ask for the same price; WISP 1 asks for a much lower price (3 times less

than for the others). The results show that by decreasing the price it is asking for, WISP 1 can "reintegrate" the network and get the clients.

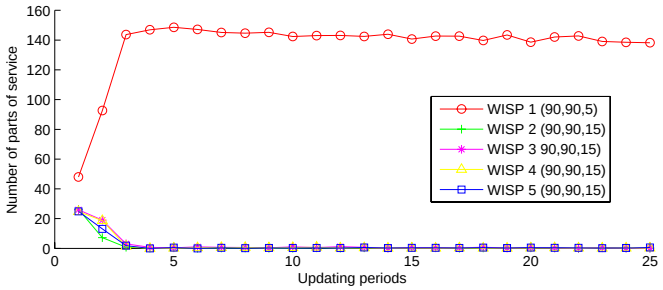


Figure 2.13: Scenario 3.3: WISP 1 has, when it opens its service, a lower reputation than for the other WISPs but it asks for much lower price; it eventually gets all the clients.

Note that even if according to the results WISP 1 gets almost all the clients, it is not interesting for it to keep the price very low because it will probably not cover its expenses; lowering the prices can therefore be considered a way of "launching" (if the initial reputation is not $maxRep$) or of "redemption" (if the WISP damaged its own reputation because it misbehaved).

2.4.4 Prediction of the QoS offered by the WISP

The results of Scenario 1 show that the WISPs are encouraged to be honest. However, this requires each WISP to accurately "predict" the QoS it can offer to its clients. This prediction depends on several parameters such as the number of neighboring WISPs, the number of clients that are simultaneously connected in the neighborhood, the clients' arrival rate, etc.

In this section, we propose the following simple prediction mechanism that consists of three main steps: (i) the estimation of the number of clients expected in the network during the next period of time, (ii) the computation of the total throughput expected in the network during the next period of time, and (iii) the definition of the prediction strategy.

Estimation of the Number of Clients

During this phase, a WISP W has to estimate, for the next period of time, the number of mobile clients that will be served in its neighborhood. This estimation has to take into consideration three main parameters:

(i) The length of the estimation period, i.e., the period of time for which the estimation is done (e.g., the next 15 minutes, the next hour).

(ii) The period of the day (e.g., peak hours, etc.) or of the year (e.g., working day, week-end, holidays, etc.) during which the estimation period is considered. This parameter gives an idea about the expected traffic.

(iii) The length of the history maintained by the WISPs. Indeed, while it operates, each WISP maintains the history of the number of clients simultaneously served in the neighborhood, the duration of the connections, the clients's arrival rate, the duration of the connections, etc. A longer history leads to a better estimation.

Computation of the Total Throughput

During this phase, W computes the total throughput expected in the network during the next estimation period. This value can be computed using the number of clients simultaneously served in the neighborhood (estimated in the previous phase) and Bianchi's throughput performance evaluation formula [18] (see Figure 2.14).

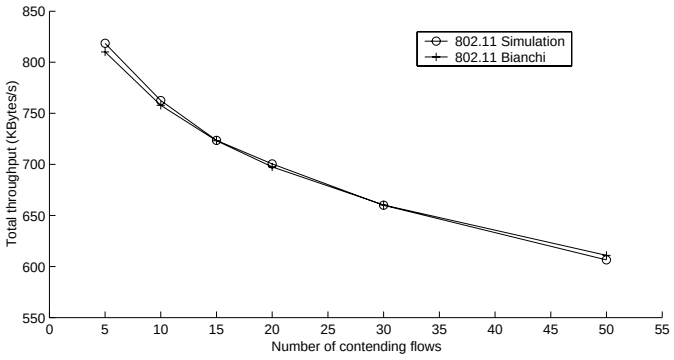


Figure 2.14: The total throughput obtained using Bianchi's throughput performance evaluation formula [18]. Bianchi's throughput is very close to the throughput we get by means of simulations.

Definition of the Prediction Strategy

Each WISP considers the value of the total throughput computed during the previous phase and decides the QoS it will advertise and to what extent it wants to "overbook" itself. The efficiency of a given strategy depends on several parameters such as the duration of the connections and the clients' arrival rate. We cannot compare these strategies as they may perform differently in different circumstances: A strategy that performs well in case of short and frequent connections may perform poorly when the connections become long and sporadic. Therefore, the WISPs may consider using different strategies according to the situation.

2.5 Analysis of the Security Mechanism

In this section, we analyze the robustness of our protocols against various attacks against our protocols (see Subsection 2.2.2 for the trust and adversarial model). We identify eight attacks that are specifically targeted against our solution: *Publicity*, *Selective Publicity*, *Denigration*, *Flattering*, *Report Dropping*, *Service Interruption*, *Refusal to Pay* and *Repudiation* attacks. We also consider the general attacks described in Subsection 1.3.2.

2.5.1 Specific Attacks

Publicity Attack: In this attack, S advertises a QoS that is higher than the real QoS it can offer. As a reaction, *MN* will send a negative report to *TCA* at the end of the session. If this attack is repeated, the cumulation of the negative reports will affect the future reputation records of S. If on the contrary, this attack is performed rarely, it will not affect much the reputation of S but S gains almost nothing from performing this attack; as S is rational, it will not perform this attack. The same reasoning holds if S=H with, in addition, the possibility for MN to punish H by choosing another home WISP.

Selective Publicity Attack: In this attack, S attempts to performs the Publicity attack with a specific *MN*. However, the anonymity of the mobile nodes prevents S (if $S \neq H$) from performing the Publicity attack against a specific *MN*. The only possible selection would be based on the home network (i.e., S performs the Publicity attack with all the *MN*s affiliated with a given home network). S gains nothing from this attack and thus S will not perform it.

Denigration Attack: In this attack, *MN* receives a good QoS from S but pretends the contrary by sending a negative report or no report at all.

If no report is sent, H will not give *MN* the ε reward and *TCA* will consider the absence of feedback as negative feedback. Therefore, this attack is not rational for

MN. Therefore, it is more interesting for *MN* to send a negative feedback instead of not sending the report at all: The effect of the attack is the same and at least *MN* will get paid for the feedback. But this attack is still not rational. Indeed, *MN* gains nothing from sending a negative feedback instead of a positive one (the cost of the sending remains the same). Such behavior is thus purely malicious.

This attack is not harmful for the WISP, unless it is performed systematically and by a high number of colluding *MN*s. This attack is rational if the *MN*s belong to a competitor that wants to affect the WISP's reputation. However, *TCA* can statistically detect it if the following events happen frequently:

1. The *MN*s affiliated with H always pretend that they received a bad QoS (i.e., lower than the advertised QoS) from a given WISP, whereas many other *MNs* report on a good QoS from that very WISP. As the selective publicity attack is not possible, this situation is suspect and TCA may punish H, e.g., by downgrading its reputation record.

2. *TCA* never receives reports from *MN*s affiliated with H about the sessions they established with S.

Note that this attack comes with an important cost: if an attacker $\mathcal{M}$ wants to alter the reputation of S by parking misbehaving nodes close to the hot spots managed by S, $\mathcal{M}$ should own many devices and devote them to the attack. Note also that this colluding attack may harm very small WISPs (with few hot spots) - if the attacker pays the price - but it is much too costly against WISPs with hundreds of hot spots.

Flattering Attack: In this attack, *MN* sends systematically a good feedback about S's behavior to *TCA*. This attack makes sense particularly if $S = H$; it significantly improves the reputation of the targeted WISP only if it is performed systematically and by a high number of colluding attackers. The detection mechanism can be similar to the one proposed for the denigration attack. However, a specificity of this attack resides in the fact that H can create "virtual" *MN*s (i.e., *MN*s that have an account but are not necessarily real devices), emulate connections with them and make them systematically send positive feedback. This leads to a cost that is much lower than the cost of the denigration attack but *TCA* can detect it if (i) the *MN*s affiliated with H rarely connect to foreign WISPs (or at least much less than average) or if (ii) H is not rewarded for the connections it established with a high number of *MN*s affiliated with it (if we assume that this information is available to *TCA*).

Report Dropping Attack: In this attack, *MN* sends the report but S does not transmit it to *TCA* (e.g., because S expects a negative feedback). However, as the absence of feedback counts as the lowest possible feedback, this dropping does not help S: Assuming that the feedback is defined between values *minRep* and *maxRep*, not receiving the report corresponds to a feedback of *minRep*. This attack

is therefore not rational for S.

Service Interruption Attack: In this attack, S receives the i-th payment proof from *MN* but does not provide the corresponding part of the service. *MN* will then keep asking for it (by sending again the i-th payment). After a predefined number of retransmission requests, *MN* will end the session, which prevents S from providing more service parts (and thus earning more money) and also affects the satisfaction level of *MN*. If nevertheless, we want to prevent S from receiving the i-th payment without providing the i-th service, we can use the solution proposed in [23].

Refusal to Pay Attack: In this attack, *MN* does not send the i-th payment to S. Then, S will not provide the i-th part of the service and the session will end (after a predefined number of retransmission requests). This attack is then not rational: It prevents *MN* from receiving the service part but does not harm S.

Repudiation Attack: In this attack, S or *MN* retracts the agreement it has with other party (e.g., S asks for higher price than agreed upon when the contract C was established). This attack is not efficient because H and *TCA* receive the contract C from both *MN* and S (Messages 2.8 and 2.10). The two copies should be identical, otherwise *TCA* will not send the message 2.11 and the session setup will not terminate. Therefore, once the session is established, *MN* and S cannot retract their agreement. To prevent S or *MN* from sending incorrect information to H, we can also require a response from H to establish the session.

2.5.2 General Attacks

Packet Dropping Attack: In this attack, $\mathcal{A}$ drops a message it is asked to forward or discards a message it is asked to generate and send. If this is done during session setup, the secure session will not be established. If $\mathcal{A}= MN$ (i.e., MN does not generate messages 2.1, 2.3, 2.7 or 2.9), it will not be able to connect to the Internet but does not harm S. If $\mathcal{A}= S$, it will not provide the service to *MN*; *MN* will select another WISP and S would lose an opportunity for revenue.

If during the secure session, the payment proof or the part of the service is not generated or is dropped, the entity that is waiting for it asks for retransmissions (if needed several times). If it does not receive the message, the session is closed.

If S does not forward the message *Sl* of *MN*, it is equivalent to the denigration attack (see Subsection 2.5.1).

If S does not generate the payment request and sends it to *H* (Message 2.15), it will not get rewarded for the service parts it provided to *MN*.

Filtering Attack: In this attack, $\mathcal{A}$ modifies a packet it is asked to forward or generate. However, the messages exchanged between the different parties in our protocols are cryptographically protected, using MAC computations or digital sig-

natures. Therefore, any modification of a message will be detected at the receiver. Therefore, tampering with a message is equivalent to not sending the message at all (an incorrect message is discarded) and it is treated in the same way (see the *Packet dropping* attack).

Replay Attack: In this attack, $\mathcal{A}$ replays a valid message that was exchanged between two parties.

During session setup, the messages exchanged between the different entities (Messages (2.2) to (2.6)) are protected using nonces; delayed messages are easily detected and discarded.

During the secure session: the payment proofs and the parts of the service arrive in sequence; a replay is immediately detected and discarded.

During session closing, the payment request and the satisfaction level (Messages (2.15, (2.17) and (2.18)) are expected only once; a replay is immediately detected and discarded.

Emulation and Node Duplication Attacks: In the emulation attack, $\mathcal{A}$ uses the secret data of a valid node. $\mathcal{A}$ can thus successfully impersonate this node during all the phases of the protocol. However, if both $\mathcal{A}$ and the legitimate node are connected to the network, this attack can be detected by H (e.g., the node seems to be at two different locations at the same time) and H can provide the legitimate node with new secret data. The identification of the legitimate node may however require human involvement.

The detection of the node duplication attack can be done in the same way.

Denial of Service Attack: In this attack, $\mathcal{A}$ prevents two or more nodes from communicating, e.g., by jamming the wireless channel. This attack is malicious and solving it may require human involvement (e.g., S identifies the jamming device and, if possible, removes it).

Intrusion Attack: In this attack, $\mathcal{A}$ is an unauthorized node but it manages to be accepted in the network as a valid node. This attack is not possible against our system. Indeed, if $\mathcal{A}$ is an unauthorized node, the authentication of $\mathcal{A}$ by H would fail and $\mathcal{A}$ would never be served by the WISP.

Sybil Attack: In this attack, $\mathcal{A}$ has different identities. If $\mathcal{A}$ wants to use all these identities, it has to register each of them with a valid home WISP H. However, in order for $\mathcal{A}$ to perform powerful attacks, it has to own a large number of identities, which leads with a significant cost for $\mathcal{A}$.

2.6 Overhead

In this subsection, we evaluate the computation and communication overhead of our solution for a mobile node. We consider only the mobile node because it

is the only entity that is severely ressource restrained and because in this way we address all the wireless aspects of the communications.

2.6.1 Computation Overhead

During the different phases of our protocols, we use symmetric and public key cryptography primitives to secure the message exchange and to authenticate the different parties involved in the communication. We minimize however the use of public key cryptography, especially by the mobile nodes, to reduce the computation cost.

Hence, *MN* uses public key primitives only for two messages: it verifies the certificate, the signature and the reputation of the WISP it selects (Message 2.2) and it encrypts a message for *TCA* (Message 2.9). For all other messages, *MN* uses symmetric cryptography primitives: $5 + 2\ell$ MAC operations (ℓ being the total number of service parts), 2 encryptions and 1 decryption.

Public key operations are also used in the message exchange between *TCA* and the two WISPs S and H (Messages 2.11, 2.18 and 2.19). It is however possible to convert them into symmetric key operations, if we assume that S and *TCA* establish a symmetric key when they first begin their interaction.

Note that the existence of a tamperproof hardware at *MN* is not necessary for the good functioning of our protocols, but it may be a good solution for protecting the long term symmetric key k_{HM}.

2.6.2 Communication Overhead

Table 2.1 provides reasonable values of the size of the different fields appearing in our protocol.

Field Name	ReqID	IDs	N_X,pad	w_i	ℓ
Size (bytes)	4	16	20	20	2
Field Name	MAC	PK	QoS, P, R	k	tag
Size (bytes)	16	150	1	16	6

Table 2.1: Size of the fields used in our protocol

ReqID is encoded on 4 bytes to reduce the risk of using the same identifier for two different requests. The identifiers of the WISPs and the nodes (W, H, S, *MN* and *TID*) are 16 bytes long (assuming an IPv6 format for example). The paywords w_i are 20 bytes long (e.g., assuming SHA) and the QoS (*AQ* and *QoSEval*), the reputation *R* and the price *P* are encoded on 1 byte each (which is enough to encode

values between 0 and 100). The symmetric keys k_{HM}, k_{HS}, k_{MS} and k_{MT} are 16 bytes long (128 bits) and the public keys are 150 bytes long (e.g., assuming RSA, see [61]). We encode the nonces (N_M and N_W) and the pads on 20 bytes, the *tag* on 6 bytes (see Subsection 2.3.2) and the MACs on 16 bytes. Finally, we use a sequence number ℓ that is 2 bytes long to support long sessions.

We consider the example where *MN* is downloading a 1 MB file. We assume that the file is divided into 1 KB packets and each 50 packets represent a part of service ($\ell = 20$ parts of service in total). Using the values of Table 4.1, an end-to-end session between *MN* and S represents an overhead, for *MN*, of 18337 bytes, which represents an overhead per packet of around 18 bytes (i.e., less than 2% of the packet size).

2.7 State of the Art

Reputation-based Systems: These systems are mainly used to build trust and foster cooperation among a given community. The efficiency of reputation mechanisms has been widely studied in various fields and with different approaches. Studies such as [41], [74] and [75] consider the effect of *online* reputation systems [30] on e-marketing and trading communities such as eBay. Reputation mechanisms are also used to foster cooperation in peer-to-peer networks [31] or in ad-hoc networks [22, 66].

But, from all these studies, we cannot draw a clear conclusion about the efficiency of reputation systems; each of these mechanisms should thus be analyzed on a per-case basis.

Roaming in WISPs: The deployment and success of WiFi networks is slowed down by the lack of interoperability between WiFi providers (this is also called the *fragmentation* problem [70]): A client that has an account with a WISP A cannot connect to a hot spot managed by a WISP B. This situation, however, is changing and more and more WISPs are establishing roaming agreements (similar to what is done for cellular networks). The roaming can be between providers within the same country (e.g., T-Mobile and iPass in the US) or on international scale (e.g., between the British BT and the American Airpath).

Another solution would be to use the service of a *WiFi roaming operator* such as *Boingo Wireless* [42]. Such an operator tries to solve the roaming problem by having agreements with as many WISPs as possible. It then aggregates all the hot spots managed by these WISPs into a single (seamless) network. However, Boingo does not consider the problem of the variable QoS in WiFi networks.

In [72], Patel and Crowcroft propose a ticket based system that allows mobile users to connect to foreign service providers: The user contacts a *ticket server* to

acquire a ticket, requests a service from a *service server* and uses the ticket to pay for that service. However, unlike the solution we present in this work, the authors of [72] do not question the honesty of the service providers, i.e. they assume that the service providers provide the users with a good quality of service, which is far from being guaranteed in WiFi networks. The same problem exists in the solution proposed by Zhang et al. [88].

In [32], Efstathiou and Polyzos present a Peer-to-Peer Wireless Network Confederation (P2PWNC) where the roaming problem is considered as a peer-to-peer resource sharing problem. They propose a solution where a WISP has to allow the foreign users to access its hot spots in order to allow its own users to connect to foreign WISPs' hot spots. This solution however presents the same problem as for [72], i.e., there is no guarantee of a good QoS provision.

2.8 Conclusion

Wi-Fi networks have a very strong potential: They are easy to deploy, they use unlicensed frequencies and they provide fast Internet connectivity. However, two major problems still need to be solved: the lack of a seamless roaming scheme and the variable quality of service experienced by the users. The reputation-based solution presented in this chapter solves both problems: It allows a mobile node to connect to a foreign Wireless Internet Service Provider (WISP) in a secure way while preserving its anonymity and it encourages the WISPs to provide the users with good QoS. Our solution takes into account the fact that the mobile clients are resource restrained mobile device and therefore have much less computing and storage resources than *TCA*, H or S.

We have analyzed the robustness of our solution against different attacks and we have shown that our protocols thwart rational attacks, detect malicious attacks and can help identify the attacker.

We have shown, by means of simulations, that the WISPs are encouraged to provide the MNs with a good QoS and, at the same time, discouraged from advertising a QoS that is different from that they can really offer.

Publications:[15, 16, 17]

Chapter 3

Ensuring fairness in Mesh Networks

3.1 Introduction

Wireless mesh networks (WMNs) have the potential to provide ubiquitous high-speed wireless Internet connectivity at a fraction of the costs of a fiber-based network. WMNs consist of Transit Access Points (TAPs) that offer Internet connectivity to the mobile clients within their communication range. A subset of these TAPs are directly connected to the Internet (e.g., using a wired connection) and can thus send and receive Internet traffic directly; we refer to these as Wired Access Points (WAPs). Regular (unconnected) TAPs have to rely on multi-hop communication to connect to the most appropriate WAP (e.g., the closest WAP in terms of hops or the least loaded WAP) and therefore to reach the Internet (see Figure 3.1).

WMNs, however, are not yet ready for wide-scale deployment due to two main reasons. First, the communications being wireless and therefore prone to interference, WMNs present severe capacity and delay constraints [35]. Nevertheless, there are reasons to believe that technology will be able to overcome this problem, e.g., by using multi-radio and multi-channel TAPs [59]. The second reason that slows down the deployment of WMNs is the lack of security guarantees. In this chapter, we first identify the security challenges introduced by WMNs by analyzing the specifics of this new kind of networks. This analysis leads us to the identification of three fundamental network operations that need to be secured: (i) the routing protocol, (ii) the detection of corrupt TAPs and (iii) the enforcement of a proper fairness metric in WMNs. We then focus on the fairness problem and propose FAME, a scheduling algorithm that ensures a fair resource allocation in WMNs. In order to reduce the complexity of the fair schedule computation, we

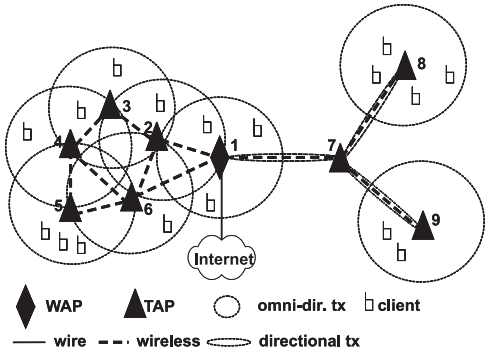


Figure 3.1: Wireless Mesh Network: The coverage of the Wired Access Points (WAP) is extended by deploying several Transit Access Points (TAPs).

develop a greedy heuristic that approximates the optimal fair resource allocation.

We evaluate FAME by means of simulations under realistic traffic loads. We compare the FAME heuristic solution to the optimal solution and observe that FAME's fair allocation is close to the theoretical optimum. This holds for constantly backlogged traffic, as well as for Web-like traffic patterns.

Finally, we experimentally assess the fairness properties on the Magnets testbed where TAPs are equipped with multiple network interfaces and directional antennas [56]. Our multi-hop measurements show that the use of directional antennas and the ability of the TAPs to send and receive concurrently alleviate unfairness but do not eradicate it; a scheduling algorithm is thus still required to ensure a fair and efficient resource usage. We implemented a simplified version of FAME that throttles flows at the WAP. Our results show significant fairness improvements.

Outline The remainder of the chapter is organized as follows. We identify the security challenges in WMNs in Section 3.2. The details of FAME are presented in Section 3.3. Section 3.4 evaluates the effectiveness of FAME via simulations and via experiments on the Magnets testbed. After discussing the security of WMNs in Section 3.5 and the state of the art in Section 3.6, we summarize the results in Section 3.7.

3.2 Security Challenges of WMNs

In this Section, we identify the security challenges introduced by WMNs by analyzing the specifics of this new kind of networks. This analysis leads us to the identification of three fundamental operations that have to be secured in WMNs.

3.2.1 Characteristics of WMNs

WMNs represent a new network concept and therefore introduce new security specifics. Here, we describe these specifics by giving an overview of the fundamental differences between WMNs and two well-established infrastructure-based technologies: cellular networks and the Internet.

Difference between WMNs and Cellular Networks The major difference between WMNs and cellular networks, beside the use of different frequency bands (WMNs usually make use of unlicensed frequencies), concerns the network configuration: In cellular networks, a given area is divided into cells and each cell is under the control of a base station. Each base station handles a certain number of mobile clients that are in its immediate vicinity (i.e., communication between the mobile clients and the base station is single-hop) and it plays an important role in the functioning of the cellular network; the entity that plays an equivalent role in WMNs would be the WAP.

However, whereas all the security aspects can be successfully handled by the base station in cellular networks, it is risky to rely only on the WAP to secure a WMN, given that the communications in WMNs are multi-hop. Indeed, centralizing all security operations at the WAP would delay attack detection and treatment and therefore would give the adversary an undeniable advantage. Furthermore, multi-hopping makes routing in WMNs a very important and necessary functionality of the network; and like all critical operations, an adversary may be tempted to attack it. The routing mechanism must thus be secured.

Multi-hopping has also an important effect on the network utilization and performance. Indeed, if the WMN is not well-designed, a TAP that is several hops away from the WAP would receive a much lower bandwidth share than a TAP that is next to it. This leads to severe unfairness problems, and even starvation [35]; it thus can be used by an adversary to disturb the functioning of the WMN.

Note that multi-hopping is also the main difference between WMNs and WiFi networks, which means that the security problems we already identified and that are related to multi-hop communications are the main security challenges introduces by WMNs, in comparison with WiFi networks.

Difference between WMNs and the Internet In WMNs, the wireless TAPs play the role that is played, in the classic (wired) Internet, by the routers. Given

that wireless communications are vulnerable to passive attacks such as eavesdropping, as well as to active attacks such as Denial of Service (DoS), WMNs are subject to all these attacks whose effects are amplified by the multi-hop aspect of the communications.

Another fundamental difference between the Internet and WMNs is that, unlike Internet routers, the TAPs are not physically protected. Indeed, they are most often in locations that are accessible to potential adversaries, e.g., deployed on rooftops or attached to streetlights. Furthermore, one very important requirement regarding the TAPs - for the concept of mesh networks to remain economically viable - is their low cost that excludes the possibility of strong hardware protection of the devices (e.g., detection of pressure, voltage, or temperature changes) [3]. The absence of physical protection of the devices makes WMNs vulnerable to some serious attacks such as tampering, capture or replication of TAPs.

This brief analysis of the characteristics of WMNs clearly shows that, compared with other networking technologies, the new security challenges are mainly due to the multi-hop wireless communications and to the fact that the TAPs are not physically protected. Multi-hopping delays the detection and treatment of the attacks, makes routing a critical network service and may lead to severe unfairness between the TAPs, whereas the physical exposure of the TAPs allows an adversary to capture, clone or tamper with these devices.

3.2.2 Three Fundamental Security Operations

Our study of WMNs' specifics led to three critical security challenges: (i) detection of corrupt TAPs, (ii) securing the routing mechanism, and (iii) definition of a proper fairness metric to ensure a certain level of fairness in the WMN. These challenges are not the only important ones that should be considered because other network functionalities also need to be secured (e.g., MAC protocols, nodes' location, etc.). We choose to focus however on these three challenges as they are, in our opinion, the most critical for WMNs.

Detection of Corrupt TAPs

As explained previously, mesh networks typically employ low-cost devices that cannot be protected against removal, tampering or replication. An adversary can thus capture a TAP and tamper with it. Note that if the device can be remotely managed, the adversary does not even need to physically capture the TAP: A distant hacking into the device would work perfectly. The WAP plays a special role in

the WMN and may handle or store critical cryptographic data (e.g., temporary symmetric keys shared with the mobile clients, long-term symmetric keys shared with the TAPs, etc.); therefore, we assume that the WAP is physically protected.

We identify four main attacks that may be performed on a compromised device, depending on the goals the adversary wants to achieve: The first attack consists in the simple removal or replacement of the TAP in order to modify the network topology to the benefit of the adversary. This attack can be detected by the WAP or by the neighboring TAPs when a brutal and permanent topology change is observed in the network.

The second attack consists in accessing the internal state of the captured device without changing it. The detection of this passive attack is difficult, given that no state change is operated on the TAP; disconnecting the device from the WMN may not be required for the adversary to successfully perform the attack; and even if a disconnection were required, the "absence" of the device may not be detected, as it can be assimilated to some congestion problem. If this attack is successful, it guarantees to the adversary the control of the corrupt TAP and a perfect analysis of the traffic going through it. This attack is more serious than simple eavesdropping on the radio channel in the sense that the adversary, by capturing the TAP, can retrieve its secret data (e.g., its public/private key pair, the symmetric key shared with the neighboring TAPs or with the WAP, etc.) and can use these data to compromise, at least locally, the security of the WMN, especially data confidentiality and integrity, and clients' anonymity. Unfortunately, there is no obvious way to detect this attack. However, a possible solution that mitigates its effect is a periodic erasure and reprogramming of the TAPs; the adversary is then obliged to compromise the device again.

In the third attack, the adversary modifies the internal state of the TAP such as the configuration parameters, the secret data, etc. The purpose of this attack can be, for example, to modify the routing algorithm at the captured node in order to change the network topology. This attack can be detected by the WAP using a verifier such as the one presented in [77] or using a solution such as the one presented in [82].

Finally, the fourth attack consists in cloning the captured device and installing replicas at some strategically chosen locations in the mesh network, which allows the adversary to inject false data or to disconnect parts of the WMN. This attack can seriously disrupt the routing mechanism, but it can be detected using the mechanism introduced in [71].

Secure Multi-hop Routing

By attacking the routing mechanism, an adversary can modify the network topology and therefore affect the operation of the network. The attack can be *rational* or *malicious*; For example, a malicious adversary may want to partition the network or to isolate a given TAP or a given geographic region, whereas a rational adversary may want to force the traffic through a specific TAP in the network (e.g., through a TAP that it has compromised) in order to monitor the traffic of a given mobile client or a given region. Another example would be for the adversary to artificially lengthen the routes between the WAP and the TAPs, which would seriously affect the performance of the network. This attack can be rational if it is performed against a competitor for example.

To attack the routing mechanism, the adversary can (i) tamper with the routing messages, (ii) modify the state of one or several TAPs in the network, (iii) use replicated node(s), or (iv) perform DoS attacks:

(i) To prevent attacks against the routing messages, the operator can use one of the existing secure routing protocols for wireless multi-hop networks [69, 47, 49, 87, 48].

(ii) If the adversary chooses to modify the state of one or several TAPs in the network, the attack can be detected (e.g., using [77] or [82]) and the operator can reconfigure the WMN accordingly.

(iii) If the adversary uses replicated node(s), the attack can be detected as the operator will realize that the network topology is not the one it originally deployed; it can therefore disable the rogue devices or install new ones [71].

(iv) Finally, DoS attacks represent a simple and efficient way to attack routing. These attacks are very harmful as they are simple to perpetrate and hard to prevent. Indeed, the adversary can disturb the communications between the TAPs in a given area and force the reconfiguration of the network. In order to solve this problem, the operator has to identify the source of disturbance [84] and, if possible, disable it.

Note that, except for the first attack, solving all these attacks requires human involvement (i.e., to go to the field and install/remove TAPs or jamming devices), which may be considered a successful attack as such.

Ensuring Fairness

In WMNs, all the TAPs use the same WAP as a relay to and from the infrastructure and therefore the throughput obtained by the TAPs can vary significantly

depending on their position in the WMN. Indeed, as shown in [35], the TAPs that are more than two hops away from WAP may starve (i.e., their clients are not able to send or receive traffic), which is highly unfair. The study conducted in [35] identifies the problem and proposes a solution that guarantees a TAP-fair share of the bandwidth. However, a TAP-based fairness is not necessarily the best solution for WMNs. Indeed, consider as an example the one-dimensional WMN presented in Figure 3.2; a per-TAP fairness policy leads to flows 1, 2 and 3 having each the same share of the bandwidth, without taking into consideration the number of clients that are served by each of these TAPs. We believe that the bandwidth sharing should be fair client-wise, because the purpose of a mesh network is to offer a service (typically Internet connectivity) to the mobile clients that are usually paying the same flat rate. That is why, in the example of Figure 3.2, flow 2 should have half as much as what flow 1 and flow 3 have, as TAP2 is serving only one client, whereas TAPs 1 and 3 are serving two clients each.

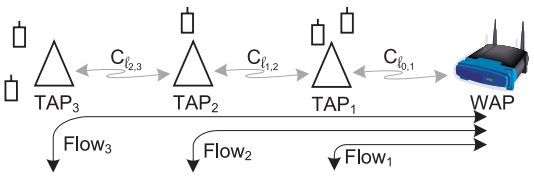


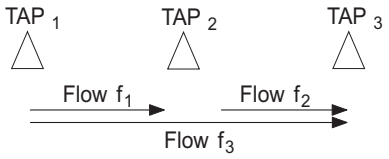
Figure 3.2: An example of a linear WMN. Each TAP serves a different number of mobile clients.

The fairness issue is closely related to the number of hops between the TAPs and the WAP; this means that if the adversary manages to increase the number of hops between a given TAP and the WAP, it can decrease dramatically the bandwidth share of this TAP. A possible solution against this attack can be a periodic reconfiguration of the WMN; given that the WAP and the TAPs are static, the operator can define - based on the traffic in the WMN - the optimal configuration of the WMN and force the routes at the TAPs to the optimal routes.

3.2.3 Studying Unfairness in WMNs

This section identifies the fairness challenges in WMNs that motivate the need for a fair scheduling algorithm.

To illustrate that bandwidth use is unfair even in a simple multi-hop topology, we set up, in our lab, the three-TAPs topology depicted in Figure 3.3 (a). All three TAPs are equipped with a single IEEE 802.11a WiFi interface with a raw data rate of 54 Mbps and omni-directional antennas. The use of the 5 GHz frequency of



(a) Simple multi-hop topology

	TCP throughput (Mbps)		
	f_1	f_2	f_3
Flows (individually)	19.1	19.1	11.4
Flows (concurrently)	5.8	5.7	2.3

(b) Throughput of each flow

Figure 3.3: Illustration of the unfairness in a WMN

IEEE 802.11a ensures that the transmissions do not interfere with the coexisting operational networks. The TAPs themselves are all in transmission range of each other, but we set up the routing such that traffic from TAP_1 to TAP_3 is routed via TAP_2. To study fairness, we compare the throughput of 3 TCP flows: Single-hop flows f_1 and f_2 are from TAP_1 to TAP_2 and from TAP_2 to TAP_3 respectively, whereas flow f_3 is a two-hops flow from TAP_1 to TAP_3. Traffic is generated with `iperf` for 3 minutes.

Figure 3.3 (b) shows the achieved throughput when each flow is activated individually, and when the three flows are activated at the same time. If f_1 is the only active flow, it can take advantage of the full effective bandwidth (19.1 Mbps). The same holds for f_2. However, if f_3 is the only active flow, it gets significantly less bandwidth (11.4 Mbps) than the single-hop flows because TAP_2 cannot send and receive at the same time.

If the three flows are active at the same time, the throughput difference between the single-hop flows and the 2-hop flow increases by a factor of 2.5. These results confirm those reported by Gambiroza et al. [35] for a different protocol and a different MAC schema. Moreover, the unfairness is likely to increase in real WMNs where the topology imposes more complex interference patterns and where flows may traverse more than 2 hops.

The unfairness illustrated above leads to three challenges that have to be addressed to achieve fairness in WMNs. First, an appropriate definition of the fairness metric is needed for WMNs: We need a metric that guarantees every user a fair share of the bandwidth, independently of its location within the WMN. In the previous experiment, if TAP3 is the WAP, a mobile client located at TAP2 would get more than twice the throughput of a client attached to TAP1. In a real deploy-

ment, a user expects to receive the same service from the operator of a WMN, independently of its location and of the distribution of the other users among the TAPs. Existing fairness definitions, such as proportional fairness (as used in TCP) or TAP-fairness [35], do not meet these constraints. In particular, TAP-fairness assumes that the same bandwidth should be provided *independently* of the number of users; such a policy is not suitable for an operator.

Second, the topology and the inferred interference pattern must be taken into account. Previous work, such as [35], only considers linear topologies. Our fair scheduling algorithm generalizes those concepts and does not depend on the WMN being organized on a specific topology such as a line or a tree.

Third, TAPs can and will be equipped with a variety of hardware. TAPs may contain a single WiFi interface or have multiple interfaces that allow concurrent sending and receiving of data. Moreover, antennas can be omni-directional or directional (directional antennas limit interference); A scheduling mechanism for fair resource allocation in WMNs has to be flexible to accommodate the hardware diversity.

3.3 FAME: FAir MEsh Scheduler

Our response to the above challenges is FAME, a novel FAir MEsh scheduling algorithm that computes a collision-free schedule based on the topology of the WMN. The schedule assigns bandwidth shares to the flow of each mobile client on the communication links this flow traverses. This schedule takes into consideration the distribution of the mobile clients in the WMN and the traffic they generate. Transmissions that do not interfere are scheduled in parallel to maximize *spatial reuse* (i.e., the possibility for links that do not contend to be activated at the same time) and thus to optimize network utilization.

In order to define the location-independent per-client fair schedule, FAME first identifies the bottleneck link of the WMN, i.e., the link with the maximum traffic-to-capacity ratio (e.g., if in the simple linear WMN depicted in Figure 3.2 we have the same capacity at all the links, the bottleneck link would be the link between TAP_1 and the WAP). Then, FAME provides equal bandwidth shares to all flows crossing the bottleneck link. The results of the fair resource allocation on the bottleneck link are then used to define the WMN allocation vector (i.e., the bandwidth share of each flow on each link of the WMN). Given that the mobile clients can create several flows, we propose that fairness is provided at the granularity of the aggregated flows per client, i.e., all data sent and received by a client. Therefore, the mobile clients cannot gain more bandwidth by generating several flows in parallel.

To address interference among TAPs in a complex WMN topology, FAME models the network as a directed graph where the TAPs represent the vertices and an edge $\ell_{a,b}$ denotes that TAP_a and TAP_b are within transmission range of each other. However, finding the optimal collision-free fair schedule in such case is an NP-hard problem, even in the simplified case where all links in the WMN have the same constant capacity and where all traffic is backlogged [13]. In this chapter, we consider WMNs where the links have different capacities and where the clients generate realistic traffic patterns (e.g., Web traffic that includes non-backlogged traffic). We then use FAME to define a heuristic that dynamically adapts the fair schedule to the traffic fluctuations.

For the sake of simplicity, we consider upstream and downstream links separately and we assume, in the remainder of this chapter, that only one WAP exists in the WMN. We also assume that the TAPs are (at least) loosely synchronized, so that each TAP can send its traffic during the time allotted to it in the schedule without interfering with the other nodes in the WMN. If we assume that the TAPs can exchange information about the distribution of the clients in the network (e.g., over a control channel), then the TAPs can periodically use FAME (see Subsection 3.3.3 for more details about the schedule update frequency) to define the fair share of the network resources as a function of the network topology, the link capacities and the number of active clients. The time is divided into cycles and FAME returns a fair collision-free schedule that assigns, to each active client in the WMN, time slots that are dedicated to its traffic during the cycle. The remainder of this section provides the details of the fair scheduling mechanism.

3.3.1 System Model and Notation

We represent the mesh network as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where the set of vertices $\mathcal{V}= \{TAP_i, i \in \{0..n\}\}$ with $TAP_0 = WAP$ and the set of edges $\mathcal{E} = \{\ell_{a,b}, a, b \in \{0..n\}\}$; $\ell_{a,b} \in \mathcal{E}$ means that TAP_a and TAP_b are within transmission range of each other. The communication link $\ell_{a,b}$ has a capacity $C_{\ell_{a,b}}$ and is *upstream* if it is used to handle the traffic from the mobile clients to the WAP and *downstream* if it is used to handle the traffic from the WAP to the mobile clients. We will denote by $\mathcal{L}_U$ the set of upstream communication links and by $\mathcal{L}_D$ the set of downstream communication links. The set of mobile clients is denoted by $\mathcal{M}= \{M_i, i \in \{0..n_M\}\}$. We assume, for the sake of simplicity, that all the nodes in the WMN (i.e., the WAP and all the TAPs in the WMN) are under the control of a single operator and that the network topology is fixed and known to all the nodes. The traffic generated and received by a mobile client M_i ($1 \leq i \leq n_M$) is represented by flows f_i^u and f_i^d, respectively. The analysis we give in this chapter is valid for both upstream and downstream traffic. Therefore, we refer to the flows

f_i^u and f_i^d using the generic notation f_i. Similarly, we refer to $\mathcal{L}_\mathcal{U}$ or $\mathcal{L}_\mathcal{D}$ using the generic notation $\mathcal{L}$.

3.3.2 FAME Design

When the WMN is first deployed, the network operator provides the TAPs with an estimate of the clients' distribution. The TAPs use this information and the network topology as an input for FAME, to define the initial fair schedule. Then, this schedule is updated using the effective distribution of clients and their traffic demands. The computation of the fair schedule consists of three main steps: (i) the construction of the cliques, (ii) the computation of the link activation times and (iii) the construction of the fair schedule.

Construction of the Cliques

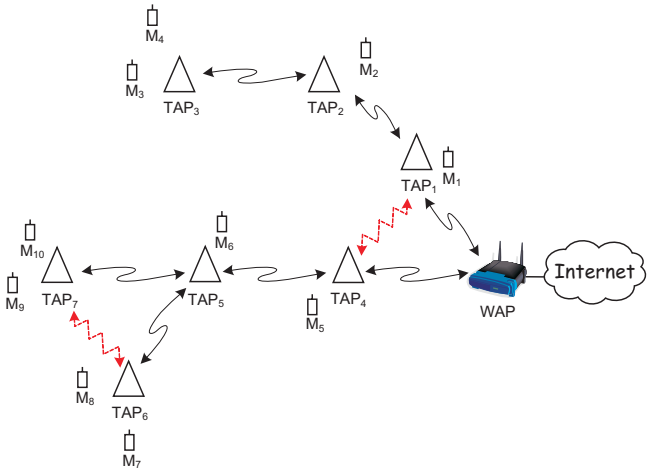


Figure 3.4: A Wireless Mesh Network (WMN) comprised of one Wired Access Point (WAP), 7 transient access points (TAPs) and 10 mobile clients (Ms). The WAP is directly connected to the Internet whereas the TAPs have to rely on wireless links to get Internet connectivity. The solid arrows represent communication links and the dashed arrows represent undesired interference.

Denote by $t_{\ell_{a,b}}$ and $t_{\ell_{a,b}}^{f_i}$ the duration of the activation of link $\ell_{a,b}$ during the cycle and the time dedicated to flow f_i on link $\ell_{a,b}$, respectively. We also denote by $F_{\ell_{a,b}}$ the set of traffic flows traversing link $\ell_{a,b}$, by r_i the route from the TAP serving M_i to the WAP, and by T the duration of the cycle.

We define the *compatibility matrix* CM as the binary matrix that indicates which links can be activated at the same time [68], i.e., that *do not* interfere:

$$CM = [cm_{x,y}], \qquad 1 \leq x, y \leq |\mathcal{L}|$$

where $|\mathcal{L}|$ denotes the cardinality of the set of links $\mathcal{L}$. We assume that all links in $\mathcal{L}$ are sorted according to a certain order and we assume that the x'th and y'th links in the sorted $\mathcal{L}$ correspond to links ℓ_{a_1,b_1} and ℓ_{a_2,b_2}, respectively. Therefore, we have:

$$cm_{x,y} = \begin{cases} 0 & \text{if } x = y \\ 0 & \text{if links } \ell_{a_1,b_1} \text{ and } \ell_{a_2,b_2} \text{ contend} \\ 1 & \text{otherwise} \end{cases}$$

For the WMN of Figure 3.4, the upstream compatibility matrix is:

$$\text{CM} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{3.1}$$

where the rows correspond to links $\ell_{1,0}$, $\ell_{2,1}$, $\ell_{3,2}$, $\ell_{4,0}$, $\ell_{5,4}$, $\ell_{6,5}$ and $\ell_{7,5}$, respectively. In the definition of this compatibility matrix, we have considered a WMN where each TAP has a single interface, i.e., each TAP cannot (i) transmit and receive at the same time, (ii) receive from more than one TAP at a time, and (iii) send to more than one TAP at a time. We have also considered a WMN where the TAPs use omnidirectional antennas. Note that the construction of the compatibility matrix is independent of the traffic model used in the WMN.

The compatibility matrix can be represented as a graph which we call the *compatibility graph* and where the vertices correspond to the links in $\mathcal{L}$. If the x'th and y'th positions in $\mathcal{L}$ correspond to links ℓ_{a_1,b_1} and ℓ_{a_2,b_2} respectively, there is an edge between vertices ℓ_{a_1,b_1} and ℓ_{a_2,b_2} if $cm_{x,y} = 1$.

We define a *clique* as a set of links that can *all* be enabled at the same time. In the compatibility graph, a *clique* corresponds to a clique, i.e., a complete subgraph. Several cliques of cardinality $c > 0$ can exist for the same WMN; we denote by Cl_c^k the k-th clique of cardinality c. In the compatibility graph constructed previously, Cl_c^k corresponds to:

- The vertex $\ell_{a,b}$ if $Cl_c^k = Cl_1^k = \{\ell_{a,b}\}$,

- The arc between vertices ℓ_{a_1,b_1} and ℓ_{a_2,b_2} if $Cl_c^k = Cl_2^k = \{\ell_{a_1,b_1}, \ell_{a_2,b_2}\}$), and
- A clique (i.e., a complete subgraph) composed of the vertices that are in Cl_c^k if $k > 2$.

We denote by d_c^k the time that is reserved, on the cycle, for Cl_c^k; we call it the *duration* of the clique. d_c^k corresponds to the maximum activation time among the c links in the clique:

$$d_c^k = \max_{\ell_{a,b} \in Cl_c^k} t_{\ell_{a,b}}$$

Therefore, the clique Cl_c^k generates a *gain* $g(Cl_c^k)$ where:

$$g(Cl_c^k) = \sum_{\ell_{a,b} \in Cl_c^k} t_{\ell_{a,b}} - d_c^k$$

The value of $g(Cl_c^k)$ corresponds to the cumulative time that would have been necessary to separately transmit the traffic on each of the links in Cl_c^k other than the link with the maximum activation time (i.e., no spatial reuse).

We define the set $\mathcal{CL}$ of *all* possible cliques; We will use these cliques to define the collision-free schedule. Even though the clique enumeration problem is proven to be NP-hard [55, 36], the relatively small size of the WMN and the utilization of optimized algorithms such as [27] or [80] can make the clique enumeration phase much more efficient and fast.

Defining the Link Activation Times

The fair schedule depends on the traffic traversing the WMN. To decide which combination of cliques leads to the best schedule, the value of the gain $g(Cl_c^k)$ must be evaluated for each clique Cl_c^k. This evaluation requires the knowledge of the activation time $t_{\ell_{a,b}}$ for each link $\ell_{a,b}$ in the network.

To compute these values, we identify the bottleneck link bl, i.e., the link with the maximum traffic to capacity ratio:

$$\frac{|F_{bl}|}{C_{bl}} = \max_{\ell_{a,b} \in \mathcal{L}} \frac{|F_{\ell_{a,b}}|}{C_{\ell_{a,b}}}$$

Each flow traversing bl should receive an equal share of the bandwidth. Therefore we have:

$$t_{bl}^{f_i} = \frac{t_{bl}}{|F_{bl}|}, \qquad \forall f_i \in F_{bl} \tag{3.2}$$

Moreover, in order to make sure that the data sent in flow f_i is transmitted in its entirety to the destination (i.e., the WAP for the upstream flows and the mobile client for the downstream flows) within the same cycle, we need to add the following condition:

$$t^{f_i}_{\ell_{a_1,b_1}} \cdot C_{\ell_{a_1,b_1}} = t^{f_i}_{\ell_{a_2,b_2}} \cdot C_{\ell_{a_2,b_2}}, \quad \forall \ell_{a_1,b_1}, \ell_{a_2,b_2} \in r_a \tag{3.3}$$

Condition 3.3 ensures that the network resources will not be wasted sending data that will remain trapped in the WMN and eventually cause queuing problems at the TAPs.

Based on Equation 3.2 and on Condition (3.3), the end-to-end throughput ρ_i is attributed to each flow f_i that traverses the bottleneck link bl where

$$\rho_i = \frac{t^{f_i}_{bl} \cdot C_{bl}}{T}$$

If we want to ensure per-client fairness condition, then we should have:

$$\rho_i = \rho_j, \quad \forall i, j \in \{1..n_M\} \tag{3.4}$$

Therefore, for any flow f_i in the network, the time $t^{f_i}_{\ell_{a,b}}$ dedicated to flow f_i on link $\ell_{a,b}$ is:

$$t^{f_i}_{\ell_{a,b}} = t_{bl} \cdot \frac{C_{bl}}{|F_{bl}| \cdot C_{\ell_{a,b}}}, \qquad \forall \ell_{a,b} \in \mathcal{L}$$

Hence, the duration of the activation of link $\ell_{a,b}$ is

$$t_{\ell_{a,b}} = \sum_{f_i \in F_{\ell_{a,b}}} t^{f_i}_{\ell_{a,b}} = t_{bl} \cdot \frac{|F_{\ell_{a,b}}| \cdot C_{bl}}{|F_{bl}| \cdot C_{\ell_{a,b}}}, \quad \forall \ell_{a,b} \in \mathcal{L} \tag{3.5}$$

Definition of the Fair Schedule

We define a schedule s as a set of cliques that fulfills the following condition:

$$\bigcup_{Cl \in s} Cl = \mathcal{L} \tag{3.6}$$

Condition (3.6) guarantees that all the links are activated at least once during the cycle. The set $\mathcal{S}$ of all possible schedules is derived from the list of cliques obtained during the *Clique Construction* phase. For each element s in $\mathcal{S}$, we define the cycle duration T_s and the gain g_s as

$$T_s = \sum_{Cl_c^k \in s} d_c^k$$

and

$$g_s = \sum_{Cl_c^k \in s} g(Cl_c^k)$$

The *network throughput* Γ is

$$\Gamma = \sum_{i=1}^{|F|} \rho_i = t_{bl} \cdot \frac{|F| \cdot C_{bl}}{|F_{bl}|} \tag{3.7}$$

where F represents the set of flows in the WMN. Of all these schedules, we need to identify the schedule that maximizes the network utilization, i.e., maximized the value of t_{bl}.

For each schedule s, the value of t_{bl}^s that satisfies the condition $T_s = T$ is:

$$t_{bl}^s = \frac{T \cdot |F_{bl}|}{C_{bl} \cdot \alpha_s} \tag{3.8}$$

where

$$\alpha_s = \sum_{Cl \in s} \max_{\ell_{a,b} \in Cl} \frac{|F_{\ell_{a,b}}|}{C_{\ell_{a,b}}} \tag{3.9}$$

In order to maximize the network throughput Γ, we have to find the schedule s^* that minimizes α_s:

$$\alpha_{s^*} = \min_{s \in S} \sum_{Cl \in s} \alpha_s \tag{3.10}$$

Unfortunately, to find the optimal schedule s^*, all possible clique combinations fulfilling Condition (3.6) have to be considered. To reduce the complexity of this exhaustive search, the following simple greedy clique combination algorithm that approximates s^* can be used:

1. Set $CliqueSet = \mathcal{CL}$
2. While $CliqueSet \neq \emptyset$
   - Given the number and distribution of the flows in the WMN, identify the clique $\widehat{Cl}_1$ such that:

$$\alpha_{\widehat{Cl}_1} = \max_{Cl \in CliqueSet} \alpha_{Cl}$$

- Set $\hat{s} = \hat{s} \cup \{\widehat{Cl}_1\}$.
- Remove from $CliqueSet$ the cliques that have one or more common links with $\widehat{Cl}_1$.

We evaluate the efficiency of this clique combination algorithm in Section 3.4.

3.3.3 Updating the Schedule

The frequency of the updates depends on a variety of parameters such as the number of active clients at each TAP and the amount of information sent and received by these clients. Frequent updates lead to a more accurate schedule and thus to a more efficient resource use, but they are expensive in terms of message exchange (between the TAPs), which makes the adaptation of the schedule to each and every traffic fluctuation not feasible. Identifying the right updating frequency is thus a tradeoff between overhead and effectiveness. A detailed evaluation of the update frequency is beyond the scope of this work.

3.4 Evaluation of FAME

In this section, we evaluate our solution first using Matlab simulations, and then using the *Magnets* WiFi testbed.

3.4.1 Evaluation via Simulations

We implemented FAME as well as the optimal scheduler using Matlab. Both algorithms construct the cliques, compute the link activation times and compute the schedule that determines when and for how long each link is activated during the cycle. The optimal scheduler Opt computes s^* by selecting the best possible clique combinations that fulfill Condition (3.6), whereas FAME uses the heuristic presented in Subsection 3.3.2 to approximate s^*. Moreover, Opt updates the fair schedule after each cycle, while FAME updates its schedule every $N \geq 1$ cycles.

The network topology on which the algorithms operate is specified by an $n \times n$ matrix A where $A(a,b) = 0$ if TAP_a and TAP_b are not neighbors and are not interfering. $A(a,b) = C_{\ell_{a,b}}$, where $C_{\ell_{a,b}}$ is the link capacity, if there is a communication link between TAP_a and TAP_b. $A(a,b) = -1$ if TAP_a and TAP_b interfere with each other. The basic time unit for the simulations is a time slot ts. The duration of a time slot is the maximum time unit such that the time needed to send a packet across any link in the WMN is an integer multiple of ts.

We use a realistic Web traffic model by using ON-OFF model for the flows where the duration (Web page size) and the inter-arrival time of the ON periods

follow a heavy-tailed distribution. We chose a Pareto distribution with $\alpha = 1.2$ and $\alpha = 1.5$ respectively [29]. Depending on the average value of the duration and inter-arrival distributions, different traffic loads can be simulated.

Opt and FAME both take the topology matrix A, the distributions of the duration and the inter-arrival time as input. FAME additionally needs the cycle update frequency N. The result of the simulation consists of:

- The amount of data generated by the different flows at each cycle,
- The amount of data sent, at each cycle, using Opt, and
- The amount of data sent, at each cycle, using FAME.

Simulations Setup We perform our experiments for the network shown in Figure 3.5. We chose this particular topology because it makes it possible to cover a given geographic area with the minimum number of TAPs. All links have an identical capacity of one (capacity unit). The TAPs at the center of the cell-shaped areas (i.e., the WAP and TAPs 3, 4, 5, 9, 10 and 11) are serving the mobile clients that are in transmission range. For these TAPs, we fix the number of clients to $n_{Clients} = 5$, resulting in a total of 30 clients in the WMN. TAPs at the cell intersections (TAPs 1, 2, 6, 7 and 8) are relaying the traffic of the neighboring TAPs to and from the WAP.

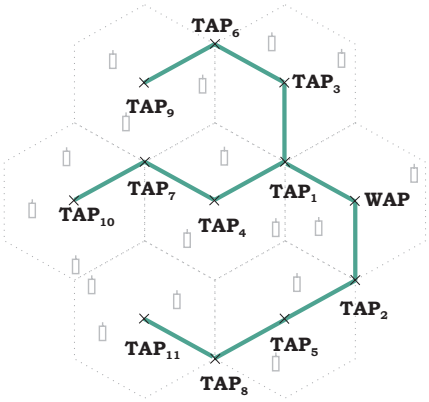


Figure 3.5: The network topology considered in the Matlab simulations.

The cycle duration is set to $T = 500\ ts$ and the total number of cycles in the simulation to 1000.

We define three traffic loads by varying the average duration of the ON-OFF model: low, moderate and high. As shown in Table 3.1, the mean OFFTime value

is the same for all three traffic loads and corresponds roughly to the duration of a cycle (i.e., 500 ts), whereas the mean ONTime value is around 10 ts, 35 ts and 49 ts for low, moderate and high traffic load, respectively. However, given that the ONTime and the OFFTime follow a heavy-tailed distribution (with $\alpha = 1.2$ and $\alpha = 1.5$, respectively), some of the values these two variables can take can be very large (see the max values in Table 3.1).

		Low	Moderate	High
ONTime	Mean	10.62	35.17	49.15
	Median	4	11	18
	Max	41951	95572	55035
OFFTime	Mean	520.01	497.52	520.47
	Median	270	270	270
	Max	1642759	181030	1488810

Table 3.1: Characteristics of the traffic models expressed in time slots (1 unit=1 ts)

Simulation Set 1: Both FAME and Opt are compared and run with the same update time, i.e., we set the schedule update time N to 1 for FAME.

Simulation Set 2: We decrease FAME's schedule update frequency by setting $N = 100$. This means that there are 10 schedule updates during the simulation.

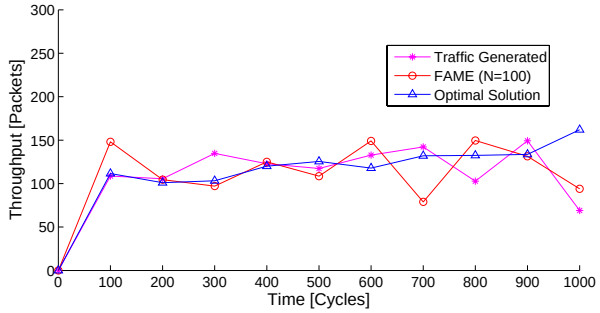
Simulation Set 3: We use the same setting as in Simulation Set 2, but assume that the TAPs use directional antennas and have several radio interfaces so that they can (i) send and receive at the same time, (ii) send to several neighbors at a time, and (iii) receive from several neighbors at a time. This results in a compatibility matrix where all values except for the diagonal ones are 1.

Simulation Results The simulation results for upstream traffic and downstream traffic are very similar. Therefore, we present only the results for downstream traffic.

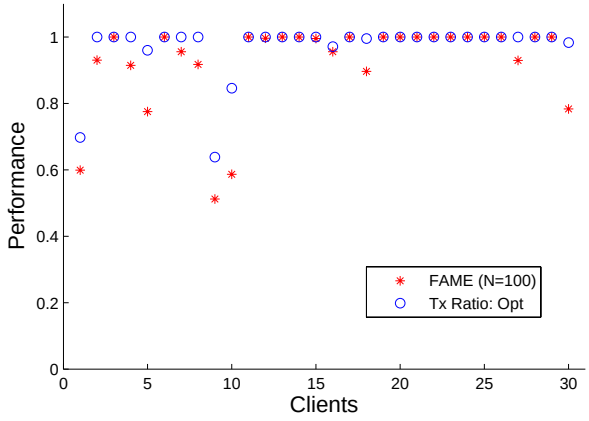
Simulation Set 1: The schedule generated by FAME coincides perfectly with the schedule generated by Opt, independently of the traffic load. These results show that $\hat{s}$ is a good approximation of s^*.

Simulation Set 2: The simulation results for the low, moderate and high traffic load are depicted in Figures 3.6, 3.7 and 3.8, respectively. Figures 3.6(a), 3.7(a) and 3.8(a) show the average throughput experienced by all clients in the WMN, the average end-to-end traffic transmitted by all clients using FAME (with $N = 100$), and the average end-to-end traffic transmitted by all clients using Opt. The x-axis denotes the time in number of cycles and the y-axis shows the throughput averaged over 100 cycles. The figures show that the difference between FAME and Opt is

small for all three traffic loads. The difference between the performance of Opt and FAME is apparent in Figures 3.6(b), 3.7(b) and 3.8(b). In these figures, we plot the ratio of sent packets vs. generated packets, of FAME and Opt for each client in the WMN. These results show that the average difference between the transmission ratio of FAME and Opt for the low, moderate and high traffic load is 0.05, 0.04 and 0.04, respectively. Therefore, we conclude that FAME is able to correctly approximate the optimal schedule even when the schedule is not updated in each cycle.

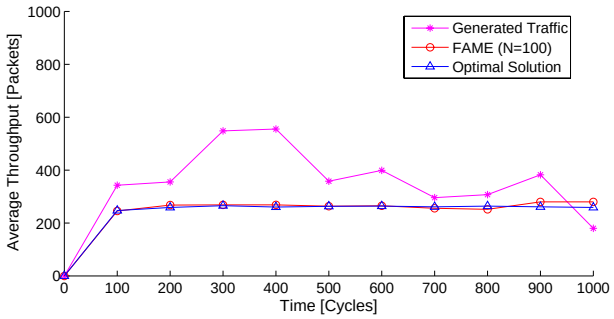


(a) Average throughput generated by the clients and transmitted using Opt and FAME (N=100).

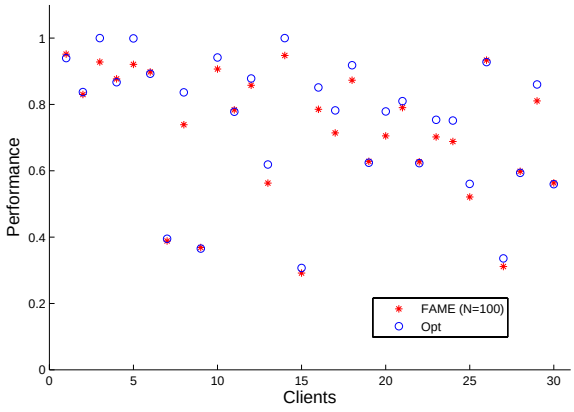


(b) Performance of FAME with N=100, compared to Opt.

Figure 3.6: Results of the Simulation Set 2 for the low traffic model.

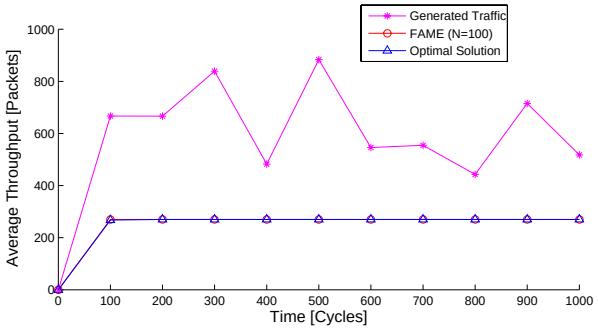


(a) Average throughput generated by the clients and transmitted using Opt and FAME (N=100).

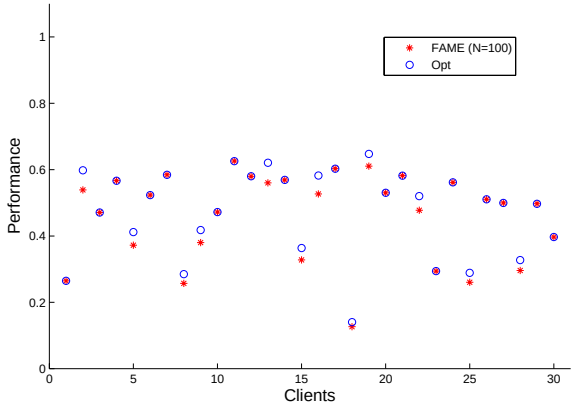


(b) Performance of FAME with N=100, compared to Opt.

Figure 3.7: Results of the Simulation Set 2 for the moderate traffic model.



(a) Average throughput generated by the clients and transmitted using Opt and FAME (N=100).



(b) Performance of FAME with N=100, compared to Opt.

Figure 3.8: Results of the Simulation Set 2 for the high traffic model.

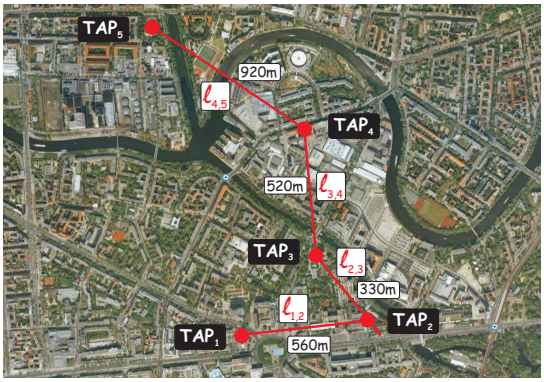
Simulation Set 3: When TAPs are equipped with directional antennas and with multiple interfaces, the maximal clique contains all the downstream links in the WMN. This is the maximal clique that is used by both Opt and FAME ($N = 100$) as a fair schedule independently of the traffic load. This means that the schedules computed by Opt and FAME are the same for all cycles. Given that this scenario corresponds to the perfect spatial reuse case, i.e., all the downstream links in the WMN can be activated at the same time, we can hardly talk about a schedule and no schedule updates are needed. Therefore, the use of FAME may be questionable. Yet, we show in the next subsection that, in practice, FAME still improves fairness in such WMNs.

3.4.2 Evaluation using the Magnets Testbed

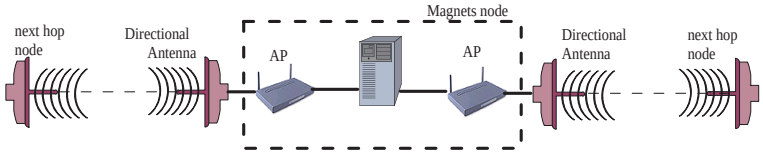
The Matlab simulation results confirm that FAME computes a good approximation of the optimal scheduler Opt. In this Section, we evaluate FAME in a wireless testbed. First, we experimentally assess the fairness properties of the Magnets wireless backbone deployed in Berlin [43] where TAPs are equipped with multiple network interfaces and directional antennas [56]. Then, we present a simplified implementation of FAME for such a wireless backbone and evaluate its fairness improvements.

Testbed Description The *Magnets* WiFi backbone is part of a metropolitan area wireless access network that is currently being deployed in Berlin. It connects five high-rise buildings using directional antennas (see Figure 3.9(a)). The distance between the buildings varies between 330m and 930m, resulting in a total span of approximately 2.3 km. All backbone components (antennas, access points) are off-the-shelf hardware. Each *Magnets* TAP consists of an IntelP4-PC based router with independent access points (APs) for each link, as depicted in Figure 3.9(b); each AP is connected to a directional antenna. All transmissions use unlicensed ISM spectrum: links $\ell_{1,2}$ and $\ell_{4,5}$ use IEEE 802.11a (5 GHz) and links $\ell_{2,3}$ and $\ell_{3,4}$ use IEEE 802.11g (2.4 GHz). For the first part of the evaluation, we focus on links $\ell_{1,2}$ to $\ell_{2,3}$ only because links $\ell_{3,4}$ and $\ell_{4,5}$ are unstable and therefore do not allow us to draw concise conclusions about the fairness properties. Then, in the second part of the evaluation, we present the measurements results for all the links in the WMN and we use them to assess throughput over four hops.

Methodology To assess the fairness properties of the backbone, we first measure the throughput characteristics of links $\ell_{1,2}$ to $\ell_{2,3}$ independently, with UDP traffic and on both directions. Given that Magnets is an outdoor testbed, the link capacities vary over time due to environmental factors [20]. Therefore we are restricted to



(a) Magnets backbone topology



(b) Magnets node

Figure 3.9: Magnets WiFi testbed in Berlin

statistical evaluations. Then, to actually assess the fairness properties, we measure the throughput of concurrent flows and multi-hop UDP and TCP flows.

We explore a total of 6 scenarios. For each of the scenarios, we run 10 experiments of 300 seconds each. We generate the traffic using `iperf` and measure the throughput at the Linux router interfaces based on packet traces gathered using `tcpdump`.

Results Scenario 1: Link Capacity Assessment.

We use Scenario 1 (see Figure 3.10) to measure the maximum capacity of each link in each direction. Accordingly, UDP traffic is injected at one end of the link and recorded at the router at the other hand and the traffic injection rate exceeds the saturation rate of each link.

Table 3.2 shows, for each flow shown in Figure 3.10, the average throughput of each link, its standard deviation, and its coefficient of variation (CoV). The CoV is the ratio of the standard deviation to the average throughput. It is an indicator of the magnitude of fluctuation. Two observations are important. First, both links have significant differences in average as well as relative variation. The throughput differs by almost a factor of 5. We attribute the throughput difference between flows $f_{1,2}$ and $f_{2,1}$ on one hand, and $f_{2,3}$ and $f_{3,2}$ on the other hand to the fact that

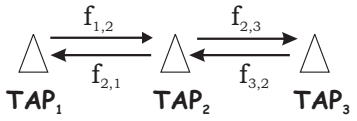


Figure 3.10: Scenario 1: Evaluation of the capacity of links $\ell_{1,2}$ and $\ell_{2,3}$, in both directions.

Flow	Avg Throughput (Mbps)	Stdev (Mbps)	Coefficient of Variation
$f_{1,2}$	31.4	1.08	0.03
$f_{2,1}$	30.6	4.44	0.15
$f_{2,3}$	6.98	1.70	0.24
$f_{3,2}$	4.81	2.00	0.41

Table 3.2: Results of Scenario 1. Each flow is activated individually.

link 1 operates in the 5GHz range and experiences less interference from neighboring networks than link 2, which operates in the 2.4 GHz range. The second observation is that the throughput of the links differs according to the direction, e.g., the average throughput of flow $f_{3,2}$ is 30% lower than the average throughput of flow $f_{2,3}$. We believe that this difference is due to a higher interference at the receiver of $f_{2,3}$ compared to the receiver of $f_{3,2}$. The same explanation holds for the difference between the throughput of flows $f_{1,2}$ and $f_{2,1}$.

Scenario 2: Two-hops communications. Next, we assess the end-to-end throughput of the 2-hops flows shown in Figure 3.11.

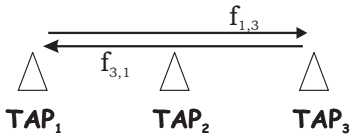


Figure 3.11: Evaluation of the end-to-end throughput of the two-hop flows f_1 and f_2.

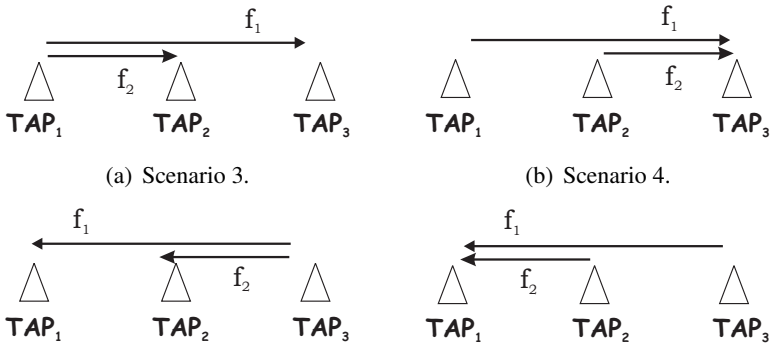
The results presented in Table 3.3 correspond to the measured throughput of the two-hop flows: In this multi-hop case, link $\ell_{2,3}$ is the bottleneck link and therefore, it provides an upper bound for the end-to-end throughput. However, the throughput should not degrade just because we consider multi-hop flows, as the TAPs are equipped with multiple WiFi interfaces and directional antennas. As such these results are different from those discussed in Section 3.3.

Flow	Average Throughput (Mbps)		Standard Deviation (Mbps)	
	UDP	TCP	UDP	TCP
$f_{1,3}$	6.99	5.27	1.48	1.43
$f_{3,1}$	4.36	6.48	2.02	1.01

Table 3.3: Results of Scenario 2. Each flow is activated individually.

The lack of fairness is visible for both TCP as well as UDP load. A direct comparison of the TCP and UDP numbers is not recommended as the raw throughput varies significantly over time; E.g., the fact that the TCP throughput of $f_{3,1}$ is higher than the UDP throughput should not be overrated.

Scenarios 3 to 6: Simultaneous Communications. Finally, we consider joint single- and multi-hop flows in four scenarios (see Figure 3.12).



(a) Scenario 3. (b) Scenario 4.

(c) Scenario 5. (d) Scenario 6.

Figure 3.12: Simultaneous communications: Scenarios 3 to 6

Figure 3.13 shows the measurements of the scenarios using UDP traffic. For each scenario, we compare the expected max-min fair allocation derived from the individual link measurements with the measured results. First, scenario 3 shows significant differences between expected and measured results, where the throughput of f_2 is much lower than expected. The reason is an unfair resource sharing on the first link: Both flows f_1 and f_2 obtain an equal share of the bandwidth (15.6 Mbps for each flow). Unfortunately, the bottleneck link for f_1 is link $\ell_{2,3}$, which has a capacity of 5.74 Mbps. Therefore almost, $1/3$ of the packets of flow f_1 are dropped at TAP_2.

For scenarios 4 and 5, a match between expected and measured results is found: The difference between the max-min fair allocation and the measured performance

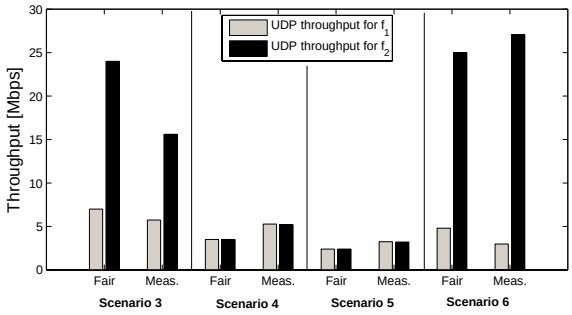


Figure 3.13: Max-min fair and measured throughput for Scenario 3 to 6 for UDP traffic.

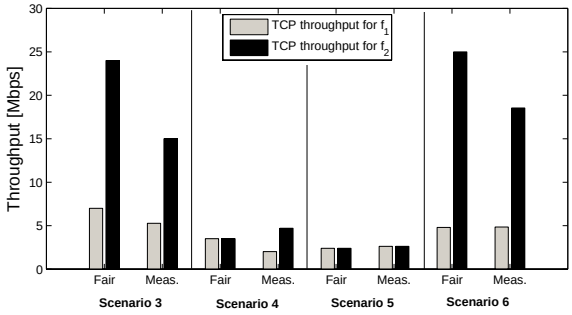


Figure 3.14: Max-min fair and measured throughput for Scenario 3 to 6 for TCP traffic.

is only due to throughput variations over time. However, the results reveal that the same problem as in Scenario 3 occurs in Scenario 4: f_1 transmits at 27 Mbps over the first link and then drops $3/4$ of its packets at TAP_2. Finally, Scenario 6 shows small differences that are attributed to bandwidth fluctuations - the results adhere to max-min fairness.

Next, Figure 3.14 shows the throughput for Scenarios 3-6 with TCP traffic instead of UDP traffic. The results are comparable to the UDP results: Only the two-hop flow f_1 in Scenario 4 shows a slight performance degradation. We believe

this degradation to be the result of bandwidth fluctuations and not fairness problems. However, no severe throughput degradation due to the multi-hop nature is visible.

These results show that the use of directional antennas and multiple interfaces alleviate unfairness, as they allow the TAPs to send and receive at the same time. However, it cannot prevent unfairness if the bottleneck link is not the first link traversed by the 2-hops flow. Therefore, the implementation of FAME can be simplified for wireless backbones: the construction of cliques can be omitted. Yet, new schedules must be calculated to account for link throughput and client distribution changes.

Implementation of FAME

In this Subsection, we describe and evaluate a prototype implementation of FAME to confirm the above conclusions. This implementation will also allow us to study the update frequency in future work.

We implement FAME as part of the Linux traffic control (*LTC*) framework. We use *LTC* to shape the outgoing traffic on the WAP. Since the WMN topology is known and static, we define a queue for each flow in LTC. Moreover, we assume that the average bandwidth for all links is known and does not change during the experiment. That is, the link bandwidth may change over time. However, since it is not possible to instantaneously update the bandwidth changes due to transmission delays between TAPs and the WAP, we ignore short-term fluctuations and focus on achieving long-term fairness. The average expected bandwidth is used to calculate the fair schedule and the results produced by the scheduler represent the input of the traffic shaper.

		Link 1	Link 2	End-to-end
without FAME	f_1	15.1	5.74	5.74
	f_2	15.6	-	15.6
with FAME	f_1	8.1	7.2	7.2
	f_2	20.8	-	20.8

Table 3.4: Throughput for Scenario 3 with and without FAME.

To evaluate FAME, we first repeat Scenario 3. Table 3.4 shows the throughput of flow f_1 and f_2 at Link 1 (T-Labs to TC) and Link 2 (TC - HHI) without and with FAME. The results for the measurements without FAME correspond to the values of Figure 3.13 and show, in addition, the throughput per link. When we use FAME to regulate the traffic, we can see that f_1 increases its throughput by 25% because f_2 is throttled at the WAP and its throughput drops by roughly 50% over

Link 1. However, this drop does not impact the end-to-end throughput as Link 2 is its bottleneck link. Therefore, these results show that FAME is able to achieve fairness.

Finally, we consider the use of FAME for the entire 4-hop topology of the Magnets backbone, resulting in flows that traverse up to 4 hops (See the scenario depicted in Figure 3.4.2). The link throughput degrades with the distance from the WAP due to environmental factors. Therefore, the need for FAME is even more acute in this scenario.

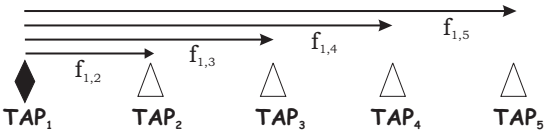


Figure 3.15: The 4-hop Magnet Backbone.

		Flow			
		f_1	f_2	f_3	f_4
without FAME	Link 1	6.82	6.33	6.45	6.60
	Link 2		4.51	4.95	4.23
	Link 3			2.30	3.13
	Link 4				3.01
	end-to-end	6.82	4.51	2.30	3.01
FAME	end-to-end	12.2	7.12	3.11	2.52
Max-Min fair	end-to-end	14	8	4	3

Table 3.5: Magnets WiFi testbed in Berlin

Table 3.5 presents the measured throughput without FAME, with FAME, and the theoretical max-min fair allocation. The results show that FAME provides a resource allocation that is close to the theoretical max-min fair allocation. The throughput throttling at the WAP does not degrade the end-to-end throughput of the flows but allows flows that cross fewer links to obtain a higher throughput.

3.5 Security Communication in WMNs

In the previous sections, we have identifies three security challenges in WMNs and focused on one of these challenges: ensuring fairness. In this Section, we will consider an example of a simple and typical communication in WMNs and we will present a simple and efficient way to secure communications in WMNs.

Figure 3.16 shows a branch of a WMN where a mobile client M is within the transmission range of TAP_3 and therefore relies on it to get Internet connectivity; the data generated and received by M goes through TAP_1, TAP_2 and the WAP. Let us consider an upstream message, i.e., a message generated by M and sent to the Internet. Before this message reaches the infrastructure, several verifications need to be performed successfully.

Figure 3.16: A typical communication in WMNs: The mobile client M is within the transmission range of TAP_3 and relies on TAP_1 and TAP_2 to relay its traffic to and from the WAP.

First of all, as Internet connectivity is a service that (usually) M has to pay for, the operator of the WMN needs to authenticate M in order to perform the billing correctly. This authentication can be done in different ways: For example, using a temporary billing account (e.g., credit card based authentication), a predefined shared secret (if M is a client of the operator managing the WMN), or a roaming system similar to the one used in cellular networks (if it is not a client of that operator); the latter has the advantage of preserving the anonymity of M with respect to a foreign operator. Note that we want to avoid, if possible, the use of asymmetric cryptographic operations by M. In fact, M being battery operated, the authentication has to be energy efficient, which makes the use of public key cryptography primitives unsuitable; these primitives have a high computational overhead and are prone to DoS attacks. Indeed, if the authentication protocol requires the computation or the verification of a signature, this feature can be misused by an adversary that can continuously ask M to compute or verify signatures; this attack can drain M's battery.

A second verification that has to be made is the mutual authentication of the network nodes (i.e., the TAPs and the WAP). We differentiate between the authentication of the nodes at the initialization (or re-initialization) phase and during the session established by M (i.e., during the sending and receiving of the packets of M).

The initialization phase takes place when the WMN is first deployed, whereas the re-initialization phase takes place when a reconfiguration of the network is needed (e.g., after the detection of attacks). The TAPs and the WAP are energy-rich and thus can use asymmetric key cryptography to perform authentication. Therefore, for the authentication of the nodes at the initialization (or re-initialization) phase, we can assume that the TAPs and the WAP have each a certified public/private key pair that is assigned to them by the operator that is managing them.

These public/private key pairs are used to mutually authenticate the nodes. This assumption is reasonable, given that the size of the WMN is relatively small and that this operation is done only occasionally. Note that M can use TAP_3's certified public key to authenticate it during session establishment.

The mutual authentication of the nodes during the session is different: the messages generated or received by M are sent using multi-hop communications and the use of public key cryptography to authenticate the sender and/or the receiver of each and every packet is a heavy process that introduces important delays and therefore leads to a suboptimal utilization of the network resources. Public key cryptography is thus not suitable in this case. Instead, the nodes can rely on symmetric key cryptography, using session keys they establish during the initialization (or re-initialization) phase or long-term shared keys that are originally loaded in the devices. If the authentication of the nodes is required at each intermediate TAP, a possible solution consists in establishing or predefining symmetric keys between neighboring TAPs; these keys would be used, typically to compute Message Authentication Codes (MACs) on the exchanged messages[1] and therefore to authenticate the nodes involved in the communication hop by hop. Otherwise, if the authentication is required only at the WAP (at TAP_3 if we are considering a downstream message, i.e., a message sent from the Internet to M), the symmetric keys can be established or predefined between each TAP and the WAP and used to compute MACs on the exchanged messages.

Once the mobile client and the nodes are authenticated, it is necessary to verify the integrity of the exchanged messages. This verification can be done end-to-end (i.e., by the WAP for upstream messages and by M for the downstream messages) or by each intermediate TAP, or both. A possible way to do this verification is for the nodes to establish a symmetric key with M at the establishment of the session; M uses this key to protect the message (e.g., using MAC). This key can also be used to encrypt the message if data confidentiality is a requirement.

3.6 State of the Art

Mesh Networks: In [4], P. Bahl et al. discuss the challenges introduced by the implementation and the deployment of public-area wireless networks (PAWNs)

[1] MACs are usually used to verify the integrity of a message, but they can also be used to authenticate the sender of the message. Indeed, assume that two parties A and B share a symmetric key k. A can generate a message m, use k to compute a MAC on it and then send both m and the corresponding MAC to B. Upon receipt of these data, B can use k to compute the MAC on m and compare it to the MAC it received; if the two MACs are identical, and given that A and B are the only two parties that know k, B can conclude that m was indeed generated by A. This authentication technique is weaker than the one that uses asymmetric key cryptography, but it is efficient.

(network security, privacy, authentication, mobility management, provisioning of key services, etc.). They describe CHOICE [5], a PAWN that they have designed and implemented. They describe the architecture and components of CHOICE, the service models it supports, and the location services and context-aware applications that they have implemented and deployed in it.

In [2], Akyildiz, Wang and Wang present a survey on recent advances and open research issues in WMNs and they point out that revising the design of MAC protocols based on TDMA or CDMA is an important research topic. Another overview of mesh networking technology is provided by Bruno, Conti and Gregori in [21].

In [25], Camp et al. present a measurement driven deployment for WMNs. The solution presented in [25] can be used during the deployment phase of the WMN in order to define judiciously the position of the TAPs in the network.

STDMA Scheduling: In [68], Nelson and Kleinrock define a broadcast channel access protocol called *spatial TDMA* (STDMA), which is designed to operate in a multi-hop packet radio environment where the location of the nodes is fixed. The defined protocol assigns transmission rights to nodes in the network in a local TDMA fashion and is collision-free. The authors propose several slot allocation methods and present an approximate solution that determines the capacity assignment for the links of the network and minimizes the average delay of messages in the system.

In [38], Gronkvist compares the *node assignment* and the *link assignment* methods. The author shows that only the connectivity of the network and the input traffic load of the network is needed in order to determine whether the node or the link assignment is preferable.

In [19] and [81], Bjorklund, Varbrand and Yuan develop mathematical formulations for resource optimization for both *node-oriented* and *link-oriented* allocation strategies. They present a column generation approach that yields optimal or near-optimal solutions. The difference with [38] is that, in [19] and [81], the authors prove the NP-hardness and present a different mathematical formulation.

Fairness in Mesh Networks: In [8], Bejerano, Han and Li propose an algorithm that determines the user-AP associations that ensure max-min fair bandwidth allocation. They study the association control problem and consider bandwidth constraints of both the wireless and backhaul links. Their formulation of the problem indicates the strong correlation between fairness and load balancing, which allows for the usage of load balancing techniques to obtain a near optimal max-min fair bandwidth allocation. Since this problem is NP-hard, they present algorithms that achieve a constant-factor approximate max-min fair bandwidth allocation.

In [35], Gambiroza, Sadeghi and Knightly study per-TAP fairness and end-to-end performance in WMNs (multi-hop wireless backhaul networks). They propose

an inter-TAP fairness algorithm that aims to achieve the per-TAP fairness objectives without modification to TCP. This work is the closest to our work, but there are a few fundamental differences:

- The definition of fairness: In [35], the authors consider a per-TAP fairness that is very well suited if a *parking lot-like* scenario[2] is considered, whereas we consider a per-client fairness that is more appropriate if we consider a WMN where all the clients pay the same monthly flat rate, which is the case we consider in this work.

- The network topology: In [35], the authors consider a single network branch, whereas we consider a network with several branches.

- The traffic model: In [35], the authors consider inter-TAP communications that do not involve the wired access point, whereas in this work, we consider that the clients are using the WMN to get Internet connectivity and therefore, we assume that the traffic is always from the clients to the WAP (upstream traffic) or from the WAP to the clients (downstream traffic).

In [13], we proposed a collision-free scheduling algorithm that ensures per-client fairness and optimizes the bandwidth utilization in a WMN where all communication links were assumed to have the same capacity C. Moreover, mobile clients were assumed to send data at saturation rate. In this work, we aim at the same objectives but consider more realistic system models: the communication links can have different and varying capacities and the generated traffic stems from a Web traffic model.

In [78], Tassiulas and Sarkar consider max-min fair allocation of bandwidth in wireless networks. In their WMN, flows that do not share nodes can transmit or receive packets at the same time. Moreover, they consider single-hop flows only. Such a system model leads to an intrinsically different formulation of the fair allocation problem and excludes many of the problems inherent to WMN and that we tackle in our work.

Jain and Das present a distributed max-min fair protocol for multihop flows in WMNs [51]. Their solution is a MAC protocol that is an extension of the VTCSMA [67] and is applied to a WMN where flows can be created between any two nodes. It is distributed but is not collision-free; it rather defines a *Recovery from Collision* protocol. FAME, in contrast, in collision-free and considers a WMN where all the traffic goes through the WAP.

[2] In the parking lot scenario, many cars attempt to leave a parking lot simultaneously using a single exit. Details can be found in [35].

3.7 Conclusion

WMNs represent a simple and inexpensive solution to extend the coverage of a WAP. However, the deployment of such networks is slowed down by the lack of security guarantees. In this chapter, we have analyzed the characteristics of WMNs and have deduced three fundamental network operations that need to be secured: (i) the detection of corrupt TAPs, (ii) the definition and use of a secure routing protocol, and (iii) the definition and enforcement of a proper fairness metric in WMNs. We have proposed some solutions to secure these operations.

This chapter presents also FAME, a FAir MEsh scheduler for wireless mesh networks (WMNs). FAME computes a fair collision-free schedule that assigns to each client in the WMN a fair share of the network resources, independently of its location in the network. We evaluate FAME via simulations and using the Magnets outdoor testbed. The simulation results show that FAME performs well compared to the Optimal fair share allocation algorithm. Moreover, the Magnet evaluations show that the use of a scheduling algorithm such as FAME is still needed even in the case when all TAPs are equipped with multiple wireless interfaces and directional antennas and fairness may be expected. We show that a simplified version of FAME suffices in this case to ensure fairness.

As future work, we intend to compare FAME to the max-min fair allocation and to perform extensive evaluation of FAME in different scenarios. In addition, we want to adapt FAME to WMNs with multiple WAPs, offering differentiated services, or controlled by several operators.

Publications:[13, 14]

Chapter 4

Cooperation in Hybrid Ad-hoc Networks

4.1 Introduction

A Hybrid Ad-hoc network is a structure-based network that is extended using multi-hop communications. Indeed, in this kind of network, the existence of a communication link between the mobile station and the base station is not required: A mobile station that has no direct connection with a base station can use other mobile stations as relays. Compared with conventional (single-hop) structure-based networks, this new generation can lead to a better use of the available spectrum and to a reduction of infrastructure costs. However, these benefits would vanish if the mobile nodes did not properly cooperate and forward packets for other nodes. In this chapter, we propose a charging and rewarding scheme to encourage the most fundamental operation, namely packet forwarding. We use "MAC layering" to reduce the space overhead in the packets and a stream cipher encryption mechanism to provide "implicit authentication" of the nodes involved in the communication. We analyze the robustness of our protocols against rational and malicious attacks. We show that - using our solution - collaboration is rational for selfish nodes. We also show that our protocols thwart rational attacks and detect malicious attacks.

The geographic area covered by a conventional structure-based network (e.g., cellular network, WiFi network, ...) is populated with base stations (also called access points) that are connected to each other via a backbone. A mobile node can use the network when it has a direct (single-hop) connection to a base station, but as soon as it is beyond the reach of the base stations' coverage, the mobile node is disconnected from the structure-based network. For the operator, the usual solution to this problem consists in increasing the coverage by adding antennas; and for the

user to move until he reaches a covered region. An alternative solution[1] would be to allow multi-hop communications in the structure-based network, which would make it possible for the isolated node to ask other nodes to relay its traffic to or from a base station.

The resulting Hybrid Ad-hoc network [1, 86, 40, 7, 83], also called *multi-hop cellular network*, offers several benefits [62, 63]. First of all, the coverage of the network is increased while the number of fixed antennas is kept relatively small. Reducing the number of antennas is beneficial for the operator because it represents a cost reduction and also because of the "NIMBY" (Not in my back yard) [58] attitude that makes site acquisition and approval both tedious and difficult. Second, the energy consumption of the nodes can be reduced because the signal has to cover a smaller distance. And finally, as the radiated energy is reduced, the interference with other nodes diminishes as well.

Given the advantages listed above, Hybrid Ad-hoc networks represent a new and promising paradigm. However, the proper operation of this new family of networks requires the mobile nodes to collaborate with each other. This collaboration cannot be taken for granted in a civilian network because each user wants to maximize his benefit while minimizing his contribution. Indeed, forwarding packets is energy-consuming and a selfish user can tamper with his mobile device to remove the relaying functions or simply shut down the device when he is not using it. A systematic denial of the packet forwarding service would remove all the benefits introduced by the multi-hop aspect of the communications.

In this chapter, we propose a set of protocols to foster cooperation for the packet forwarding service in Hybrid Ad-hoc networks. This solution is based on a charging and rewarding system.

This work extends and completes our previous treatment of the same problem [11]. This work is part of the MICS Terminodes Project [50]. The rest of the chapter is organized as follows. We introduce the system, including the adversarial model, in Section 4.2 and describe our proposed protocols in Section 4.3. In Section 4.4, we evaluate the incentive mechanism, and in Section 4.5, we analyze the robustness of our solution against rational and malicious attacks. In Section 4.6, we present an estimate of the communication and computation overhead of our protocols. Finally, we describe the related work in Section 4.7 and we present our conclusions and future work in Section 4.8.

[1] Note that we do not assume that multi-hop communication is always the best solution to increase infrastructure coverage. The decision whether or not a given network should be extended using multi-hopping is out of the scope of this work.

4.2 System and Adversarial Model

4.2.1 Assumptions

The system consists of a set of *base stations* connected to a high speed backbone and a set of *mobile nodes*. The mobile nodes use the base stations and, if necessary, the backbone to communicate with each other or with a host connected to the backbone. Communication between the mobile nodes and the base stations is based on wireless technology and the nodes are loosely synchronized with their base station. We assume that all communication is packet-based and that all the base stations and the backbone are operated by a *single operator* that is fully trusted by all mobile nodes, be it for charging, for route setup, or for packet forwarding. For the sake of simplicity, we consider that the nodes and the base stations have the same power range, which, we assume, will lead to bidirectional links (i.e., even if the quality of the link is not necessarily the same in both directions, we assume that the communication is still possible in both directions).

We call a *cell* [62] the geographical area that is controlled by a given base station. The power range of the base station is smaller than the radius of the cell, meaning that some nodes have to rely on *multi-hop relaying* to communicate with the base station. We consider a model in which the nodes move. However, we assume that the routes are stable enough to allow for the sending of a substantial number of packets and thus to amortize the cost of running a routing protocol (see Section 4.6). We assume each node i to be registered with the operator and to share a long-term symmetric key K_i with it. K_i is the only long-term cryptographic material stored in i. The secret keys of all the nodes in the network are maintained at the operator.

4.2.2 Rationale of the solution

When a mobile node A (the initiator) wants to communicate with another mobile[2] node B (the correspondent), it first establishes an *end-to-end session* with B. As we will see in detail, in Subsection 4.3.2, a session is a route on which all nodes are authenticated. This is done by establishing an *initiator session* between A and the base station of the initiator BS_A and a *correspondent session* between the base station of the correspondent BS_B and B. These sessions are used to exchange packets between A and B, in both directions.

For each packet, we call S its source (which is A or B) and D its destination (therefore B or A, respectively). The base stations of S and D are denoted by BS_S and BS_D, respectively. The packet is then sent by the source S to BS_S, if necessary

[2]We consider mobile-to-mobile communication as it is the most complete case.

in multiple hops. If D resides in a different cell, then the packet is forwarded by BS_S to BS_D via the backbone. Finally, the packet is sent to D, possibly in multiple hops again. If one of the routes is broken, then a new session is established using an alternative route. Note that the system model described above is similar to that of [62], with the difference that we require *all* communication to pass through a base station. Although this may lead to sub-optimal routes, our model has the advantage of significantly reducing the complexity of routing from the nodes' point of view, since they have to maintain only a single route (to the base station) instead of one route per correspondent. Of course, the base station has to maintain a route to every node in its cell.

To encourage the intermediate nodes to forward the traffic, we propose to charge the initiator A for the traffic in both directions and to reward the forwarding nodes (the operator is rewarded as well). We take advantage of the presence of the trusted operator and assume that it maintains a billing account for every node in the system; our remuneration scheme (see Subsection 4.3.4) is implemented by manipulating the appropriate billing accounts.

Our protocols are based entirely on symmetric key cryptography. Although asymmetric cryptographic primitives may seem to be more suitable for implementing some of the functions of our scheme, they have a high computational overhead (compared to symmetric key primitives), which prevents their application in resource constrained mobile devices.

4.2.3 Adversarial model

We do not attempt to ensure data confidentiality or node anonymity and thus, we do not study *passive* attacks (where the attacker analyzes the data without altering it). Instead, we are interested in *active* attacks, where the attacker modifies, deletes or injects data in the network. We consider the attacks described in Subsection 1.3.2.

4.2.4 Interaction with the underlying routing protocol

Our solution assumes the existence of an underlying (proactive or reactive) Ad-hoc routing protocol that provides the initiator A and the base station BS_B with the *initiator route* (route between A and BS_A) and the *correspondent route* (route between BS_B and B), respectively. The main incentive for the nodes on these routes to cooperate in the routing is the expected future benefit (i.e., the remuneration). Our solution does not require the underlying routing protocol to be secure. Indeed, the operator is able, in our solution, to detect several routing attacks such as those described in [49] (see Subsection 4.5.7 for more details).

4.3 Details of the Protocols

4.3.1 Building blocks and notation

Our protocols use two cryptographic building blocks: A MAC (Message Authentication Code) function and a stream cipher [65]. However, our use of these primitives is unconventional:

- During the session setup phase (see Subsection 4.3.2), we need all the nodes in the path to authenticate the request message and, instead of appending one MAC computed by each of the nodes to the message, we use an iterative "MAC layering" technique. The principle of this technique is explained in Subsection 4.3.2. Our solution achieves a similar effect to that of the classical MAC appending technique but keeps the size of the request constant. Therefore, our technique is more efficient in terms of bandwidth usage. To the best of our knowledge, such a scheme has not been proposed yet for Ad-hoc networks.

- During the packet sending phase (see Subsection 4.3.3), we apply an iterative stream cipher encryption mechanism that can be considered as an "implicit" authentication mechanism because it allows the operator to verify that the packet took the route it was supposed to take. At the same time, it thwarts the free-riding attack (see Subsection 4.5.2).

Notation: We denote the concatenation operator by $\mid$ and the XOR operator by $\oplus$.

4.3.2 Session setup

As explained in Section 4.2, when an initiator A wants to communicate with a correspondent B, it first has to set up an *end-to-end session*. The goal of the session setup is (i) to test the *initiator* route (route between A and BS_A, containing a relays) and the *correspondent* route (route between BS_B and B, containing b relays), obtained from the underlying routing protocol; (ii) to authenticate all nodes belonging to these routes; and (iii) to inform these nodes about the traffic that will follow. A node can decide to not join the session, in which case the session setup fails and a new session is established using an alternative route. Successful completion of the session setup phase is a confirmation that both the initiator and correspondent routes are operational and that the end-to-end intermediate nodes accept to forward the traffic.

In order to set up a session, A generates an initiator session setup request message $AReq_0$ that contains a fresh request identifier $AReqID$ (e.g., generated in se-

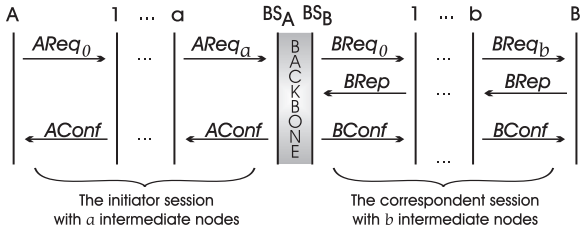


Figure 4.1: The session setup phase

quence), the initiator route $ARoute$, and some information $TrafficInfo$ about the traffic to be sent[3]. In addition, the request has a field $oldASID$ to carry the session ID of the broken initiator session, in case the request is sent to re-establish a broken session. This field is set to zero in the case of a new session establishment. Finally, $AReq_0$ contains a MAC computed by A using its secret key K_A:

$$\begin{aligned} AReq_0 \;\; = \;\; & [\, AReqID \mid oldASID \mid ARoute \mid TrafficInfo \mid \\ & MAC_{K_A}(AReqID \mid oldASID \mid ARoute \mid TrafficInfo)\,] \end{aligned}$$

Each forwarding node i ($1 \leq i \leq a$) on the initiator route checks the traffic information $TrafficInfo$. If i decides to participate in the forwarding, then it computes a MAC on the whole message using its own key K_i, replaces the MAC in the request with the newly computed MAC, and forwards the request $AReq_i$ to the next hop (or to BS_A) where:

$$AReq_i \;\; = \;\; [\, AReqID \mid oldASID \mid ARoute \mid TrafficInfo \mid MAC_{K_i}(AReq_{i-1})\,]$$

Thus, when the request arrives to BS_A, it contains a single "layered" MAC that was computed by A and all the nodes on the initiator route in an iterative manner. BS_A then repeats all the MAC computations and checks the result against the MAC in the received request. It also verifies that the request ID is fresh (i.e., the message is not a duplicate) and if the request is sent to re-establish a broken initiator session, it verifies that $oldASID$ corresponds to a valid session identifier previously initiated by A. If one of these verifications is not successful, then BS_A drops the request, otherwise it sends the request, via the backbone, to the base station BS_B. BS_B generates and sends a correspondent session setup request $BReq_0$ towards B:

$$BReq_0 \;\; = \;\; [\, BReqID \mid oldBSID \mid BRoute \mid TrafficInfo\,]$$

[3]The initiator A may not have any precise information about the traffic B will generate. $TrafficInfo$ is thus an estimate for the expected traffic in both directions. If A underestimates the traffic, the relaying nodes might interrupt the packet forwarding because the amount of data to forward is much larger than expected.

where $BReqID$ is a fresh request identifier generated by the base station BS_B, $oldBSID$ is the session ID of the broken correspondent session, in case the request is sent to re-establish a broken session and $BRoute$ is the correspondent route.

Each forwarding node j ($1 \leq j \leq b$) on the correspondent route computes and sends $BReq_j$ in the same way as the forwarding nodes in the initiator route did:

$$BReq_j \quad = \quad [\, BReqID \mid oldBSID \mid BRoute \mid \mathit{TrafficInfo} \mid MAC_{K_j}(BReq_{j-1}) \,]$$

When B receives the request $BReq_b$, it returns to BS_B a correspondent session setup reply $BRep$ that contains the correspondent request ID $BReqID$ and a MAC that is computed over the received request $BReq_b$ (including the MAC therein) using the key K_B of B:

$$BRep \quad = \quad [\, BReqID \mid MAC_{K_B}(BReq_b) \,]$$

The reply is relayed back without any modifications to BS_B on the reverse route of the request. BS_B checks the "layered" MAC and if it verifies correctly, BS_B informs BS_A that the session is valid. Then BS_A (respectively, BS_B) sends an initiator (respectively, a correspondent) session setup confirmation message towards A (respectively B). The initiator session setup confirmation message $AConf$ contains the initiator request ID $AReqID$ and two freshly generated random numbers $AUSID$ and $ADSID$ representing the initiator session IDs to be used for packets sent from A to BS_A and from BS_A to A, respectively. It also contains a series of MACs where each MAC is intended for one of the nodes on the initiator route (including A):

$$\begin{aligned} AConf \quad &= \quad [\, AReqID \mid AUSID \mid ADSID \mid AMAC_A \mid \\ &\qquad AMAC_1 \mid \ldots \mid AMAC_a \,] \\ AMAC_i \quad &= \quad MAC_{K_i}(AReqID \mid AUSID \mid ADSID \mid oldASID \mid \\ &\qquad ARoute \mid \mathit{TrafficInfo}) \end{aligned}$$

The correspondent session setup confirmation $BConf$ has a similar structure:

$$\begin{aligned} BConf \quad &= \quad [\, BReqID \mid BUSID \mid BDSID \mid \\ &\qquad BMAC_1 \mid \ldots \mid BMAC_b \mid BMAC_B \,] \\ BMAC_j \quad &= \quad MAC_{K_j}(BReqID \mid BUSID \mid BDSID \mid oldBSID \mid \\ &\qquad BRoute \mid \mathit{TrafficInfo}) \end{aligned}$$

Each node on the initiator and correspondent routes (including A and B) verifies its own AMAC or BMAC and stores the two initiator or correspondent session

IDs, respectively. The state information related to the established sessions (including session IDs, routes and cryptographic parameters) is stored in the operator's database. Then, using its secret key K_i and the session identifier, each node i involved in the communication generates a session key K_i' (e.g., $K_i' = h_{K_i}(SID)$, $SID = AUSID$ and $ADSID$ if i is in the initiator route, and $SID = BUSID$ and $BDSID$ if i is in the corresponding route, which leads to two session keys for each node, one for each direction of the communication) that it will use during the packet sending and the payment redemption phases. The base stations BS_A and BS_B also compute the session keys of all the nodes involved in the communication and save them locally.

The session becomes active for the base stations when they send the confirmation messages and for the nodes when they receive a valid confirmation message. Node i starts a timer t_i when it receives the request message; t_i is restarted each time i receives a valid message or packet that belongs to the session. Node i closes the session if t_i expires; closing a session means that the node discards all subsequent messages or packets that belong to the session. The nodes and the base stations keep state information in the memory until the acknowledgement and (if needed) packet receipts are sent to the operator (see Subsection 4.3.4).

Note that in the case of initiator (respectively, correspondent) session re-establishment, it is not necessary to also re-establish the correspondent (respectively, the initiator) session if the latter is still valid. The broken session is re-established using an alternative route and it is linked to the other (still valid) session in the operator's database.

4.3.3 Packet sending

Once the session has been set up, S (which is A or B) starts sending packets to the destination D.

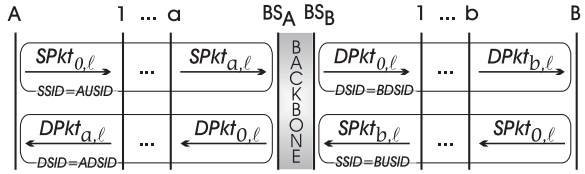


Figure 4.2: The packet sending phase

The ℓ-th packet $SPkt_{0,\ell}$ sent by S contains the session ID $SSID$ (which is called $AUSID$ if $S = A$ and $BUSID$ if $S = B$), the sequence number ℓ, and the payload $Payload_\ell$. It also contains the "receipt seed" $SRcpt_{0,\ell}$ (details about the

computation and the use of the receipts are given in Subsections 4.3.4 and 4.3.4). In addition, S computes a MAC on the packet using the session key K'_S and encrypts the body of the packet (including the MAC) by XORing it with the pad $PAD_{S,\ell}$:

$$\begin{aligned} & SPkt_{0,\ell} &=& [\, SSID \mid SRcpt_{0,\ell} \mid \ell \mid Body_{0,\ell} \,] \\ \text{where} \quad & SRcpt_{0,\ell} &=& MAC_{K'_S}(SSID \mid \ell) \\ \text{and} \quad & Body_{0,\ell} &=& PAD_{S,\ell} \oplus [\, Payload_\ell \mid MAC_{K'_S}(SSID \mid \ell \mid Payload_\ell) \,] \end{aligned}$$

The pads $PAD_{i,\ell}$ are generated by node i ($i = S$ for the source) as follows (see Figure 4.3): The session ID $SSID$ ($DSID$ for the down-stream nodes) and K'_i are used as a seed to initialize the key stream generator of the stream cipher. Then, $PAD_{i,\ell}$ is chosen as the ℓ-th block of length $MaxLength$ of the generated key stream, where $MaxLength$ denotes the maximum allowed length of packets in bytes. If the length L_ℓ of the packet to be encrypted is smaller than $MaxLength$, then only the last L_ℓ bytes of $PAD_{i,\ell}$ are used, the rest of $PAD_{i,\ell}$ is thrown away.

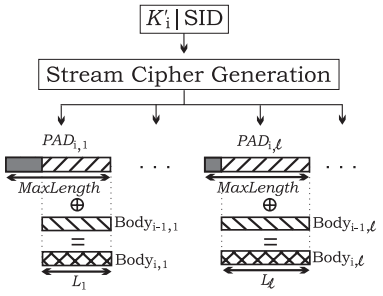


Figure 4.3: Encryption of the packets

The node i in the up-stream route (route between S and BS_S) verifies that the packet is not a duplicate, updates (and stores) the receipt[4] $SRcpt_{i,\ell}$ (details are in Subsection 4.3.4) and encrypts the body of the packet using the pad $PAD_{i,\ell}$:

$$\begin{aligned} & SPkt_{i,\ell} &=& [\, SSID \mid SRcpt_{i,\ell} \mid \ell \mid Body_{i,\ell} \,] \\ \text{where} \quad & SRcpt_{i,\ell} &=& MAC_{K'_i}(SSID \mid SRcpt_{i-1,\ell}) \\ \text{and} \quad & Body_{i,\ell} &=& PAD_{i,\ell} \oplus Body_{i-1,\ell} \end{aligned}$$

When BS_S receives the packet, it retrieves the session keys of the nodes on the up-stream route, recomputes the pads and removes all encryptions from the packet. If the resulting packet verifies correctly (i.e., it is not a duplicate and it has

[4]The receipt $SRcpt_{i,\ell}$ can be used by node i as a proof that it correctly received the packet $SPkt_{i,\ell}$ (see Subsection 4.3.4 for more details).

a valid MAC), the packet is forwarded[5] to the base station of the destination BS_D, otherwise it is dropped. BS_D changes the up-stream session ID to the corresponding down-stream session ID $DSID$ (which is $BDSID$ if $S = A$ and $ADSID$ if $S = B$), computes a new MAC for D, computes the pad $PAD_{j,\ell}$ for each node j on the down-stream route (route between BS_D and D), including D, and encrypts the packet (including the MAC) by iteratively XORing it with all these pads. The result is:

$$\begin{aligned} DPkt_{0,\ell} &= [\, DSID \mid \ell \mid Body_{0,\ell} \,] \qquad \text{where} \\ Body_{0,\ell} &= PAD_{1,\ell} \oplus \ldots \oplus PAD_{d,\ell} \oplus PAD_{D,\ell} \\ &\quad \oplus [\, Payload_\ell \mid MAC_{K'_D}(DSID \mid \ell \mid Payload_\ell) \,] \end{aligned}$$

BS_D stores $MAC_{K'_D}(DSID \mid \ell \mid Payload_\ell)$ of every packet it sends together with the sequence number ℓ in order to be able to verify future destination acknowledgements and packet receipts. Note that for the down stream, we do not need to add a field dedicated to the receipt; the receipt is generated using several fields of the down-stream packet (see Subsection 4.3.4).

Upon reception of $DPkt_{j-1,\ell}$, each node j computes and stores the receipt $DRcpt_{j,\ell}$ for the packet (as explained in Subsection 4.3.4), decrypts the body of $DPkt_{j-1,\ell}$ by XORing it with the pad $PAD_{j,\ell}$, and forwards the result $DPkt_{j,\ell}$ to the next hop where:

$$DPkt_{j,\ell} = [\, DSID \mid \ell \mid Body_{j,\ell} \,] \qquad \text{and} \qquad Body_{j,\ell} = PAD_{j,\ell} \oplus Body_{j-1,\ell}$$

When the packet reaches D, it removes the remaining encryption pad by XORing the packet with $PAD_{D,\ell}$. D can then verify the validity of the MAC generated by BS_D and store the MAC and ℓ for the generation of the acknowledgement (see Subsection 4.3.4). Note that for up-stream and down-stream packets, removing the encryptions and verifying the correctness of the resulting packet implicitly identifies the forwarding nodes and ensures that the packet took the right route.

4.3.4 Payment Redemption

Charging

As we have already mentioned in Subsection 4.2.2, charging and remuneration are performed by the network operator, by manipulating the accounts of the nodes. When BS_S receives the packet Pkt_ℓ of length L_ℓ sent by the source S, the up-stream forwarding nodes are credited $\alpha(L_\ell)$ and the initiator A is charged

[5] The packet is forwarded only if it is a data packet. The treatment of up-stream acknowledgement packets is presented in Subsection 4.3.4.

$n(L_\ell)$. Both $\alpha(L_\ell)$ and $n(L_\ell)$ depend on the packet size and not on the number of forwarding nodes in the path. The operator will then take a loss for long routes but will make a profit from short routes. The charges and rewards should thus be set so that – relative to the average path length – the operator makes the desired profit.

The down-stream forwarding nodes are credited when Pkt_ℓ is acknowledged by D (see Subsection 4.3.4) because the operator may have no other reliable information about the delivery of the packet. The only incentive for D to not send the acknowledgement is to save resources. In order to discourage this misbehavior, D is charged a small amount ε when BS_D injects Pkt_ℓ in the down-stream route and is reimbursed when Pkt_ℓ is acknowledged. Note that, as the operator cannot distinguish between a packet loss and the case where D does not want to send the acknowledgment, it keeps the charge ε if no acknowledgement arrives for Pkt_ℓ.

If the packet is dropped or lost in the up-stream route, the nodes that relayed it can present the receipt for this packet (see Subsection 4.3.4) to the operator. The operator identifies the last node k ($1 \leq k \leq u$) in the path who sent a valid receipt for the packet and gives it a reward $\beta(L_{min})$, whereas the nodes that are before k in the path receive a reward $\alpha(L_{min})$, where L_{min} denotes the minimum length of a packet. This choice of reward is made because if the reward is higher than $\alpha(L_{min})$, the forwarding nodes may be tempted to drop short packets in order to get higher rewards than the ones they would get if they forward them. A is charged $n'(L_{min}) = (k-1) \cdot \alpha(L_{min}) + \beta(L_{min})$. Receiving $\beta(L_{min})$ can be perceived by k as its reward for informing the operator that the nodes 1 to $k-1$ in the path behaved properly. The β-reward should be sufficiently large to strongly counterbalance the cost c of forwarding the packet and the cost c' of maintaining and sending the receipt ($\beta \gg c$ and $\beta \gg c'$). The α-reward should also be substantially larger than β ($\alpha \gg \beta$) to prevent nodes from systematically dropping packets. Note that even if c and c' are not constants (e.g., they depend on the battery level of the node), we can choose the α and β-reward in such a way that the conditions listed above are fulfilled.

If the packet is dropped or lost in the down-stream route, the nodes that relayed it are rewarded in a similar way as for the up-stream forwarding nodes, except for $\alpha(L_{min})$ and $\beta(L_{min})$ that are replaced by $\alpha(L_\ell)$ and $\beta(L_\ell)$, respectively, because the operator received the packet and knows its real length L_ℓ. The initiator A is fully charged $n(L_\ell)$.

Destination acknowledgement

The destination D must acknowledge every packet it correctly receives. However, in order to save resources, it does not send acknowledgements on a per packet basis. Instead, the session is subdivided into "time periods" and the packets re-

ceived during each period are acknowledged in a single batch. The acknowledgment $DAck_t$ of the t-th time period of the session is formatted as the payload of a regular packet[6] and sent by D via the down-stream route to BS_D:

$$\begin{aligned} DAck_t &= [\, Batch_t \mid DFPkt_t \mid DLPkt_t \mid DLost_t \mid \\ &\qquad MAC_{K'_D}(Batch_t \mid DFPkt_t \mid DLPkt_t \mid DLost_t) \,] \end{aligned}$$

where $DFPkt_t$ and $DLPkt_t$ are the sequence numbers of, respectively, the first and the last received packets during the t-th time period, $DLost_t$ is the list of the missing packets between $DFPkt_t$ and $DLPkt_t$ and

$$Batch_t = \bigoplus_{DFPkt_t \leq \ell \leq DLPkt_t;\ \ell \notin DLost_t} MAC_{K'_D}(DSID \mid \ell \mid Payload_\ell)$$

where $MAC_{K'_D}(DSID \mid \ell \mid Payload_\ell)$ is the MAC received in the packet Pkt_ℓ.

The packet is forwarded as a regular packet of the session. When BS_D receives it, the packet is decrypted and identified as being an acknowledgement. Then, BS_D verifies the MAC and checks $Batch_t$ by XORing all the MACs of the packets from $DFPkt_t$ to $DLPkt_t$, excluding those in $DLost_t$ and comparing the result with the received value. If the verification fails, then BS_D ignores the acknowledgement. If BS_D does not receive $DAck_t$ during the $t+1$-th time period or if the throughput is not satisfactory (i.e., too many lost packets), an alternative route is used to establish a new session.

Up-stream acknowledgment

To attenuate the effect of several malicious attacks (see Section 4.5), the base station BS_S sends a single acknowledgment $UAck_t$ to S for all the packets it received during the t-th time period of the session. $UAck_t$ is sent in a regular packet and its format is similar to the format of $DAck_t$, except that the base station does not have to provide a $Batch$-like proof to the source:

$$\begin{aligned} UAck_t &= [\, UFPkt_t \mid ULPkt_t \mid ULost_t \mid \\ &\qquad MAC_{K'_S}(UFPkt_t \mid ULPkt_t \mid ULost_t) \,] \end{aligned}$$

When S receives $UAck_t$, it identifies it as being an acknowledgement and checks its validity by verifying its MAC. S can choose to re-establish the session to BS_S using an alternative route if no acknowledgement arrives for a given time period or if the throughput is unsatisfactory.

[6] It is necessary to be able to differentiate between a data packet and an acknowledgement (e.g., by using a flag bit).

Packet receipts

The concept of receipt we use in this work is similar to the one used in [89]. It does not represent a proof that the node forwarded the packet but rather that it received it correctly. As we will see in Subsection 4.4, the use of the receipts helps to make packet forwarding rational.

For an up-stream forwarding node i, the receipt $SRcpt_{i,\ell}$ for the packet Pkt_ℓ is sent together with the payload and it is computed as explained in Subsection 4.3.3. We need a field dedicated to the receipt in the up-stream part of the communication, because if a part of the packet is used to compute the receipt, BS_S has no way to verify it in the case of packet loss, which is the very purpose of the receipts. For a down-stream forwarding node j, the receipt $DRcpt_{j,\ell}$ is computed as follows: $DRcpt_{j,\ell} = MAC_{K'_j}(DSID \mid M_{j,\ell})$ where $M_{j,\ell}$ represents the MAC field of the packet $DPkt_{j,\ell}$. It is possible for the operator to verify the receipts because it stores the MACs of the packets (they are also used to compute/verify the destination acknowledgements).

In order to save memory space, both up- and down-stream forwarding nodes do not store the receipts for each packet but rather for a whole session; the forwarding node i stores a *batch* for each session it is involved in as a forwarding node: $Batch_{SID,i} = \bigoplus_{\ell \leq LPkt\ ;\ell \notin Lost} Rcpt_{i,\ell}$ where $LPkt$ is the sequence number of the last packet received so far and $Lost$ is the set of the sequence numbers of missing packets preceding $LPkt$.

Note that for a node in the initiator route, $AUSID$ and $ADSID$ correspond to two distinct sessions. When a given session is closed and the last destination acknowledgement is sent, the operator informs the forwarding nodes, typically when the node is within the power range of a base station, about the rewards they received (e.g., using a packet similar to the up-stream acknowledgement). If a node i forwarded a packet Pkt_ℓ and was not paid for it, i sends the receipt to the operator. If the receipt is valid, the node is rewarded as explained in Subsection 4.3.4. A single receipt is sent to ask remuneration for several packets:

$$\begin{aligned} Rcpt_{SID,i} \quad = \quad & [\ SID \mid Batch_{SID,i} \mid LPkt \mid Lost \mid \\ & MAC_{K'_i}(SID \mid Batch_{SID,i} \mid LPkt \mid Lost)\ ] \end{aligned}$$

Upon reception of this message, the operator verifies the MAC and if the verification is positive, it remunerates the node according to the rewarding scheme (see Subsection 4.3.4). Note that a node can ask for remuneration (by sending the receipt) even if it did not provide the service; this attack is studied in Subsection 4.4.

4.4 Analysis of the Incentive Mechanism

The aim of the incentive mechanism we propose in this chapter is to make forwarding packets for other nodes rational. In order to assess the efficiency of our solution, we analyze the Packet Dropping attack where an attacker $\mathcal{A}$, that is part of the end-to-end route between S and D, decides to drop a packet it is asked to forward. In this section, we consider the effect of the attack on the different phases of our protocols and we show that this attack is not rational. This result proves, particularly for the packet sending phase, that our solution fosters cooperation.

Session setup phase: $\mathcal{A}$ can drop one or several of the following messages:

- The request message: The sender of the request (which is A or BS_B) does not receive the confirmation or the reply message, respectively. It then establishes a new session to the target (BS_A and B, respectively) using an alternative route. Note that dropping the request message is not necessarily an attack because the forwarding nodes can decide to not participate in a given session.

- The reply message: BS_B never receives the reply and the correspondent session setup fails. It then uses another route to establish the correspondent session.

- The confirmation message: Some of the nodes involved in the communication are not aware of the establishment of the session. If the initiator A is the source of the first packet to be sent during the session, we can have two cases: (i) $\mathcal{A}$ is in the initiator route, therefore A does not receive the confirmation message and considers that the session setup failed; it then establishes a new session using another route. (ii) $\mathcal{A}$ is in the correspondent route, the session is then active for all the nodes, except for those that are after $\mathcal{A}$ in the correspondent route (including B); these nodes discard all the packets sent by A during the session. B is thus unable to send the periodic acknowledgment to BS_B and the session is re-established.

  The problem is totally symmetric if B is the source of the first packet of the session. In both cases, this attack is not rational and can be detected rapidly by the operator.

Packet sending phase: We show here that denying to forward packets is not rational; cooperation is thus the best choice for a selfish, rational node.

Proposition 1 *If a node i received a packet Pkt_ℓ to forward and if, later on, Pkt_ℓ was not acknowledged by the target (BS_S for the up-stream and D for the down-*

stream), then it is rational for i, once the session is closed, to send a receipt for Pkt_ℓ to the network operator.

Proof: As explained in Subsection 4.3.4, after a given session is closed, the operator informs the nodes involved in that session about the rewards they received. If a node i correctly forwarded (or simply received) Pkt_ℓ and was not paid for it, i can send a receipt for it.

Sending a receipt $Rcpt$ of length L_{Rcpt} (see Section 4.6 for numerical values) represents a cost of $c'/NumPkts$ per packet, where $NumPkts$ denotes the number of packets received by i during the session and c' denotes the cost of sending $Rcpt$. Given the assumption of route stability (see Subsection 4.2.1), it is possible to neglect $c'/NumPkts$ in comparison with c (and thus in comparison with α and β) because $NumPkts$ is large.

If i decides not to send a receipt for Pkt_ℓ or if it sends an invalid receipt, then its payoff is:

- 0 if i dropped Pkt_ℓ during the packet sending phase,
- $-c$ if it forwarded Pkt_ℓ but none of the following nodes sent a valid receipt for it,
- $\alpha - c$ if it forwarded the packet and at least one of the following nodes in the path sent a valid receipt for the packet.

If i sends a valid receipt for Pkt_ℓ, then its payoff is:

- β if i dropped Pkt_ℓ during the packet sending phase,
- $\beta - c$ if it forwarded Pkt_ℓ but none of the following nodes sent a valid receipt for it,
- $\alpha - c$ if it forwarded the packet and at least one of the following nodes in the path sent a valid receipt for the packet.

Given that (i) a forwarding node cannot know if the receipt is valid or not before sending it to the operator, (ii) the cost of sending the receipt is negligible and (iii) $\alpha \gg \beta \gg c$, we can state that sending the receipt is rational. □

Proposition 2 *If all the nodes involved in the communication are rational, then forwarding the packet Pkt_ℓ is rational for node i.*

Proof: As we will show in Subsection 4.5.2, the filtering attack is malicious. As the nodes involved in the communication are rational, they will not perform this attack on the packets they are asked to forward and thus the receipts produced by the intermediate nodes will be correct.

If node i decides to defect and drops a packet Pkt_ℓ it is asked to forward, i will still send a receipt for Pkt_ℓ since, according to Proposition 1, this is the rational behavior. The payoff of i would then be β.

If i decides to cooperate, then:

- If Pkt_ℓ reaches its target, then the payoff of i is $\alpha - c$,
- If, on the contrary, Pkt_ℓ does not reach its target, then at least one node j ($j > i$) will send a receipt for it (according to Proposition 1) and the payoff of i is also $\alpha - c$.

As we have $\alpha \gg \beta \gg c$, cooperation is rational for node i. □

Proposition 3 *If the route contains an attacker that repeatedly drops the packet Pkt_ℓ, then the network operator can identify it.*

Proof: As long as Pkt_ℓ is relayed by rational nodes, the packet is computed and correctly forwarded until it reaches the malicious node $\mathcal{A}$ that drops it. The rational nodes that are before $\mathcal{A}$ in the path will then send valid receipts for Pkt_ℓ (according to Proposition 1). The operator identifies the last node k in the path that sent a valid receipt, which is $\mathcal{A}$ or the rational node that is before it on the route (because $\mathcal{A}$ is also able to generate a valid receipt for the packet). The operator suspects then both k and $k + 1$ of misbehavior. By crosschecking the information about different sessions and identifying the nodes that are suspected significantly more than average, the operator can identify the attacker and punish it in consequence. Note that if $\mathcal{A}$ performed this attack only a few times, then the detection would be slower but the attack would be less harmful. □

Proposition 4 *Forwarding the packet Pkt_ℓ is rational for node i even if an attacker $\mathcal{A}$ will drop it later on.*

Proof: Node i has no information about whether the nodes after it in the path are rational or not. If it expects all of them to be rational, then the best choice for i is to cooperate (according to Proposition 2). If it expects node $i+1$ to be rational, then the best choice for i is to cooperate (its payoff would be $\alpha - c$ because according to Proposition 1, $i + 1$ would send a receipt for the packet). Finally, if it expects node $i + 1$ to be malicious and drop the packet, then the best choice for i is also to

cooperate, because otherwise the operator would eventually believe it is malicious (according to Proposition 3) and would punish it. □

Payment redemption phase: The acknowledgement is encapsulated in a regular packet and the body is encrypted by all the nodes in the path, including the generator of the acknowledgement. An attacker $\mathcal{A}$ has thus no way to distinguish a packet containing an acknowledgement from a data packet, especially if some padding is used to prevent the acknowledgement packet from having a fixed and predefined length. A brute force attack would be for $\mathcal{A}$, in order to specifically drop the t-th acknowledgement, to drop all the packets sent during the $t + 1$-th time period. The consequence of this attack is the re-establishment of the session using another route.

4.5 Security Analysis

In this section, we study the robustness of our set of protocols against the active attacks identified in Subsection 4.2.3.

4.5.1 Replay attack

We consider that a replay attack performed by an attacker $\mathcal{A}$ is successful if the replayed message or packet is considered as valid by *all* the parties involved in the communication (including the operator). Note that $\mathcal{A}$ is not necessarily part of the network. In this Subsection, we will show that this attack is malicious and never successful.

Session setup phase: The operator maintains the information about all the sessions established so far. The replayed message (request, reply or confirmation) is thus detected by the first base station that receives it. A detection at the nodes is also possible; when a node i receives a replayed request message, it can identify it as a duplicate (and discard it) if:

- i is not part of the route in the request,
- or i is supposed to be the initiator of the communication,
- or the session to be established is already active or it is closed but still in memory. Indeed, even if the mobile nodes do not keep track of all the messages and packets they received, they do maintain a short-term history (i.e., on-going sessions and session that are not acknowledged yet).

Packet sending phase: As for the session setup phase, the duplicate is detected by the first base station that receives it. But here, the intermediate nodes are also

able to detect it because each forwarding node maintains the list of all packets it has received so far (for the computation of the receipt, see Subsection 4.3.4). The sequence number of the packet to forward corresponds then to the identifier of an already handled packet and the duplicate is discarded.

Payment redemption phase: The operator maintains the list of all acknowledgements and receipts it receives and can thus detect (and discard) a replayed message. Furthermore, as explained in Subsection 4.4, it is difficult to identify the packets containing the acknowledgements and thus to replay them specifically.

4.5.2 Filtering attack

An attacker $\mathcal{A}$ that performs a filtering attack modifies one or several fields of the packet it is asked to forward. In this subsection, we analyse the effect of this attack on our protocols. We also consider the *free-riding* attack where two colluders $\mathcal{A}_1$ and $\mathcal{A}_2$, on the end-to-end route, attempt to piggyback data (using appending or substitution) on the exchanged packets, with the goal of not having to pay for the communication.

Session setup phase: $\mathcal{A}$ can tamper with:

- The request or the reply messages: The verification of the "layered" MAC fails and the base station (BS_A or BS_B) discards the message. A new session is then established using an alternative route.
- The confirmation message: The first node that receives the tampered message discards it because the verification of the MAC fails. If $\mathcal{A}$ tampers with one (or more) MAC(s) in the message, the first node whose MAC was modified and that receives the message discards it. This attack has the same effect as dropping the confirmation message (see Subsection 4.4) and is detected in the same way.

The fields of the session setup messages are not encrypted. It is then possible for two colluders $\mathcal{A}_1$ and $\mathcal{A}_2$ to piggyback information. However, the size of fields is small enough to make the sending of useful data very long and fastidious.

Packet sending phase: $\mathcal{A}$ can tamper with the different fields of the packet Pkt_ℓ.

- Modifying SID, ℓ or $Body_{i,\ell}$ is detected by the target of the packet (BS_S for the up-stream and D for the down-stream) because the "layered" MAC does not verify correctly.
- We hereafter define the *early duplicate* attack, a malicious attack where $\mathcal{A}$ creates a fake packet with a sequence number ℓ that it expects to be used by

the legitimate source in the (near) future. This packet is considered as valid by the intermediate nodes (because they cannot verify it) but it is discarded at the target because the MAC is not correct. However, when the source sends the "real" ℓ-th packet, the forwarding nodes consider it as a duplicate and thus discard it. Our protocols, as presented so far, are vulnerable to this attack. If the operator wants to attenuate the effect of this subtle attack, it can do so (at the cost of a small overhead) by making use of hash chains (i.e., a chain of N hash values where w_N is chosen at random, $w_{N-i} = h(w_{N-i+1}), 0 < i \leq N$, and h is a one-way hash function).

Let us first describe the solution for the initiator session. During the session setup phase, the base station BS_A sends the first hash values AUw_0 and ADw_0 of two sufficiently long hash chains, in the initiator confirmation message, to the nodes in the initiator route (including A). BS_A also sends the hash value AUw_m encrypted with the secret key of A in the confirmation. A can thus retrieve the elements 0 to m of the hash chain and send the hash value AUw_ℓ ($1 \leq \ell \leq m$) with the ℓ-th packet it generates[7]. BS_A sends the hash value ADw_ℓ with the ℓ-th packet it sends toward A. The intermediate nodes can verify the validity of the hash values by checking that $w_0 = h^\ell(w_\ell)$ ($w = AUw$ or ADw). The verification of the hash value can be optimized if we use mechanisms such as [28] for example. The packets containing invalid hash values are discarded.

The solution is totally symmetric for the correspondent session. Note here that given w_ℓ, one can retrieve the hash values of all the previous packets in the session. This means that packets out of order should be discarded. But this constraint is logical in our case because we use the notion of sessions. All the packets are then expected to go through the same route and to arrive in order; the contrary is thus suspicious.

The use of the hash values can also solve the case where the attacker tampers only with w_ℓ; the attack is detected at the first node that receives the modified packet because the checking of the hash value fails.

Modifying both w_ℓ and ℓ is an even more subtle malicious attack. Let us assume that a forwarding node receives the packets $Pkt_{\ell-1}$ and Pkt_ℓ to forward. It discards $Pkt_{\ell-1}$ and replaces the sequence number and the hash value in Pkt_ℓ by $\ell - 1$ and $w_{\ell-1}$, respectively. The sequence number and the hash value are considered as valid by the following forwarding nodes.

[7] When A is about to run out of hash values, the base station provides it (in the same way the up-stream acknowledgment is sent) with a hash value AUw_{m+n}. A can then compute n new valid hash values.

Of course, the packet is discarded at the target because the MAC is not correct. The attack is possible if the attacker is part of the route and thus all the nodes on the route are suspected by the operator. The first direct effect of this attack is for the source to cancel the session, because the throughput is too low; the second effect is that the operator eventually, by crosschecking the information about the suspected nodes, identifies the attacker.

- The free-riding attack is not rational during the packet sending phase; the data sent by $\mathcal{A}_1$ cannot be interpreted by $\mathcal{A}_2$ because it was encrypted at least by one intermediate node[8]. If this attack is performed anyway, it is detected as a "regular" filtering or packet dropping attack (depending on whether $\mathcal{A}_2$ forwarded the tampered packet or not).

- Modifying only the receipt $SRcpt$ in the up-stream packets (there is no field dedicated to receipts in the down-stream packets) is a malicious attack. If the base station BS_S detects such an attack (the packet is correct but the receipt is not), then it re-establishes the session (if $S = B$) or asks the initiator to do it (if $S = A$). Such a radical solution is needed because, as explained in Subsection 4.3.4, the nodes maintain one batch per session by XORing all the receipts of the packets they handled. If one of these receipts is incorrect, then the batch is incorrect and the receipt does not verify correctly at the operator.

- The attacker $\mathcal{A}$ can tamper with the packet it is asked to forward but without altering the fields used by the intermediate nodes to generate the receipts. The following nodes in the route forward the modified packet. When the target (BS_S or D) receives it, it detects the attack and re-establishes the session.

Payment redemption phase: This attack is similar to the packet dropping attack during the payment redemption phase.

4.5.3 Emulation and Node Duplication Attacks

This attack is equivalent to the cloning of a SIM card in a GSM cellular network and can be detected in the same way; a node claiming to be in several physical locations simultaneously (e.g., it is in two geographically distinct cells) is automatically suspected by the operator. Furthermore, statistical methods can be used to

[8]Having two colluding nodes that are neighbors and that perform the free-riding attack makes no sense because they can communicate directly with each other.

determine whether certain nodes relay more traffic than is reasonable, given the type of the node. Either of these events suggests that the device is dishonest.

The same analysis holds for the node duplication attack.

4.5.4 Denial of Service Attack

In this attack, $\mathcal{A}$ prevents two or more nodes from communicating, e.g., by jamming the wireless channel.. This attack is malicious and solving it may require human involvement (e.g., the operator identifies the jamming device and, if possible, removes it).

4.5.5 Intrusion Attack

In this attack, $\mathcal{A}$ is an unauthorized node but it manages to be accepted in the network as a valid node. This attack is not possible against our system. Indeed, if $\mathcal{A}$ is an unauthorized node, the authentication of $\mathcal{A}$ by the base station would fail and $\mathcal{A}$ will never be able to send or receive packets, or to be part of a route.

4.5.6 Hybrid attacks

Sophisticated attacks can combine two or more of the attacks described so far. For example, two colluders $\mathcal{A}_1$ and $\mathcal{A}_2$ that are on the same route may want to perform, respectively, the filtering attack and the packet dropping attack. If the filtering attack does not modify the information needed by the intermediate nodes to compute the receipts, the operator will detect a "regular" packet dropping attack and will identify $\mathcal{A}_2$ as being the attacker (see the proof of Proposition 3). If, on the contrary, the nodes that are between $\mathcal{A}_1$ and $\mathcal{A}_2$ are not able to generate valid receipts, then $\mathcal{A}_1$ will be identified by the operator as an attacker that performed a filtering attack (see the Appendix). The same reasoning can be applied to the case where there are more than two colluders.

4.5.7 Securing the routing protocol

As stated in Subsection 4.2.4, even if the underlying routing protocol is not secure, the operator is able to detect several routing attacks. Indeed, during the session setup, the initiator and correspondent routes are tested and the nodes belonging to these routes are authenticated, which allows the operator to detect attacks such as routing loops or invalid routes. However, some routing attacks cannot be detected before the packet sending phase (e.g., *Sybil Attack*, *Black hole* or *Gray hole* attacks [49]); the network operator can then employ statistical methods to detect them. Note that securing the routing protocol is out of the scope of this work; we

therefore consider, to exemplify, the following attacks that we believe are the most pertinent regarding our solution:

***Sybil Attack* attack:** In this attack, the adversary makes the route appear longer by adding virtual nodes [49]. The operator determines statistically if the set of intermediate nodes is inconsistent (e.g., an emulated node is in the route or an attacker is performing the wormhole attack) or if the route is much too long (a route in Hybrid Ad-hoc networks is not expected to be long, having a too long routes is therefore suspicious). The operator can also suspect such an attack if two or more nodes seem to be always neighbors, despite mobility. More heuristics can be found in [52].

***Black or gray holes* attack:** This attack is similar to the packet dropping attack during the packet sending phase.

4.6 Overhead

In this section, we estimate the communication and computation overheads of the solution we have described. Reasonable values of the size of the different fields appearing in our protocol are provided in Table 4.1. *NbFwdrs* is the number of forwarding nodes on the route (up-stream or down-stream), ℓ is the sequence number of the packet and *NbLostPkts* is the number of packets lost during the session or the time period.

Field Name	ReqID	SID	TrafficInfo	Route
Size (bytes)	4	4	16	*NbFwdrs**16
Field Name	MAC	ℓ	SRcpt	Lost
Size (bytes)	16	2	1	*NbLostPkts**2

Table 4.1: Size of the fields used in our protocol (for both up and down streams)

The request ID and the session IDs are encoded on 4 bytes each to reduce the risk of using the same identifier for two different requests or sessions. The field *Route* is the concatenation of the 16 byte identifiers (assuming e.g. an IPv6 format) of the nodes. The *TrafficInfo* field is used to inform the forwarding nodes about the traffic to be generated; using 16 bytes to encode it seems to be reasonable. Finally, we encode ℓ on 2 bytes to support long sessions and *SRcpt* on only 1 byte because its computation and storage should be lightweight.

4.6.1 Communication Overhead

Session Setup Phase: According to Table 4.1, establishing an end-to-end session with $NbFwdrs$ forwarding nodes (in each of the routes) represents an overhead of $156 + NbFwdrs * 64$ bytes.

The session setup overhead is directly related to the lifetime of the sessions, which, in turn, very much depends on the stability of the routes.

Description of the simulations: We consider a network composed of 100 nodes laid out on a 500x500 m^2 single cell and one base station situated in the center of the cell. We fix the power range of the nodes and the base station to 100 m. We use the random waypoint mobility model [54] with a 0 s pause time and we discard the first 1000 seconds of simulation time to remove the initial transient phase [26]. We perform 3 sets of simulations where the speed is uniformly chosen between x and 10 m/s, x = 2, 3 and 4 m/s [85], which corresponds to an average speed *AvrSpeed* = 5.6, 6.7 and 7.8 respectively; we run 100 simulations for each value of *AvrSpeed*. As we are interested in the lifetime of the routes and not in communication interface, our simulation is written in plain C++ instead of ns-2.

Figures of interest: In our simulations we are interested in the two following figures:

- The average lifetime of a route (*AvrLT*): After the initial transient phase of each simulation, we randomly choose a node that has a route to the local base station (we choose the shortest path, the effect of mobility on the performance of more sophisticated routing protocols is discussed in [6]) and we observe the lifetime of this route. The simulation ends when at least one link on the route is broken. *AvrLT* represents the average value of all these lifetime values over the 100 simulations.

- The average number of forwarding nodes (*NbFwdrs*): This number is computed for the node we consider for the *AvrLT*.

Results: The results, given in Table 4.2, show that the stability of the routes decreases with higher mobility of forwarding nodes. For *AvrLT*, we consider a 95% confidence interval (*CI*).

In order to estimate the amount of information that a node can send during this period of time, let us consider the case where the nodes are running a *Voice over IP* application using a G.711 Codec (Rate = 64 kbit/s) with a frame size (including the headers) of 200 bytes [37]. If we consider that the average speed = 7.8 m/s, the route remains stable for an average of 7.8 s; it is possible during this period to send 62.4 kbytes of data. The overhead of an end-to-end session setup is 252 bytes (the average number of forwarding nodes is 1.5), which represents only 0.4% of

AvrSpeed	NbFwdrs	AvrLT (s)	95% CI
5.6 m/s	1.3	10.7	2.1
6.7 m/s	1.4	8.1	1.7
7.8 m/s	1.5	7.8	1.5

Table 4.2: Simulation results for the different values of the speed (pause time=0 s)

the amount of information (payload) that is possible to send during the session. Moreover, as explained in Subsection 4.3.2, it is possible to re-establish only the broken session (the initiator session or the correspondent session), which reduces this overhead.

The presence of one (or more) active malicious attackers in the end-to-end route can also lead to a session re-establishment. However, the operator can statistically identify the attacker(s) (see Section 4.5); the risk of being identified and punished represents a disincentive to cheat.

Packet Sending Phase: Considering the field sizes of Table 4.1, we can see that the packet sending phase represents an overhead of 23 bytes for up-stream packets and 22 bytes for down-stream packets. If the packet size is 200 bytes (considering again the VoIP example), the overhead represents at most 11.5% of the packet size. This overhead is reduced if we use larger packets.

Sending the Acknowledgment: The destination acknowledgement and the up-stream acknowledgement are generated each time period and their sizes are 36+2*$NbLostPkts_t$ bytes and 20+2*$NbLostPkts_t$ bytes, respectively. The receipt $Rcpt_{SID,i}$ is a 23+2*$NbLostPkts$ bytes message that the node i sends directly (i.e., without relaying) to the operator once per session. We expect the number of packets lost to be small in both cases (i.e., acknowledgement and receipt), otherwise the session is re-established because the throughput is not satisfactory.

4.6.2 Computation Overhead

In this subsection, we consider the computation overhead for the mobile nodes. The overhead is expressed in terms of battery consumption and number of computations. However, as shown in [73], we can consider the battery consumption, due to cryptographic computations, as negligible compared to the energy needed for data transmission.

Session Setup Phase: This operation requires all the nodes to perform 1 MAC computation and 1 MAC verification each.

Packet Sending Phase: For each packet, the source and the destination have to perform one MAC operation each. However, the main overhead in this phase

is represented by the usage of stream cipher encryption (performed by the source and all the forwarders), which ensures the authentication of the nodes involved in the communication and prevents the free-riding attack. But stream ciphers are very fast, and some operate at a speed comparable to that of 32 bit CRC computation [44].

Acknowledgment computation: For the destination acknowledgement, D performs one MAC computation/time period and one XOR operation/packet. For the up-stream acknowledgement, S performs one MAC verification per time period. Finally, for the receipts, each forwarding node performs one MAC computation/time period and one XOR operation/packet.

Numerical example: As an example, a Celeron 850 MHz processor under Windows 2000 SP can perform a MAC computation (and verification) with HMAC/MD5 algorithm at 99.863 Mbytes/s and a stream cipher encryption (and decryption) using Panama Cipher (little endian) algorithm at 120.301 Mbytes/s [44]. These numbers provide an order of magnitude; if slower (or faster) processors are used, they would of course scale correspondingly.

4.7 Related work

In this section, we discuss some research efforts related to the issues of the cooperation of nodes in (pure) Ad-hoc networks and in Hybrid Ad-hoc networks.

Cooperation in Ad-hoc networks: Several research groups have considered the problem of selfishness and the stimulation of cooperation in mobile ad-hoc networks. In [34], Félegyházi et al. establish the connection between the ad-hoc network topology and the possible existence of cooperation. In [64], Marti et al. consider the case where a node agrees to cooperate but fails to do so. Their solution uses a "watchdog" mechanism to identify the misbehaving nodes and a "pathrater" mechanism to construct routes that avoid those nodes. Both the CONFIDANT [22] and the CORE [66] approaches propose a reputation based solution to identify and punish misbehaving nodes. In [89], Zhong et al. rely on a central authority that collects receipts from the forwarding nodes and charges/rewards the nodes based on these receipts. In [24], Buttyán and Hubaux use a virtual currency/credits to charge/reward the packet forwarding service provision in ad-hoc networks.

Cooperation in Hybrid Ad-hoc networks: In [60], Lamparter et al. propose a rewarding scheme to encourage cooperation in hybrid networks (i.e., mobile ad-hoc networks with access to the Internet, which they call "stub ad-hoc networks"). They assume the existence of an Internet Service Provider that authenticates the nodes involved in a given communication and takes care of charging or rewarding them. However, [60] and our current approach present two main differences. First

of all, in [60], the authors analyse the robustness of their solution only against rational attacks, whereas in our proposal we consider malicious attacks as well. The second difference is that the cryptographic functions used in [60] are based on public-key cryptography, whereas our solution is based solely on symmetric key cryptography, which is more suitable for resource constrained mobile devices.

In [53], we have proposed a micro-payment scheme for Hybrid Ad-hoc networks that encourages collaboration in packet forwarding. However, our current proposal significantly differs from [53] in many aspects. First of all, in [53], we assume an asymmetric communication model, where the up-stream communication is potentially multi-hop and the down-stream communication is *always* single-hop, whereas in this work, both the up-stream and the down-stream communications are potentially multi-hop. Second, in [53], the nodes report a fraction of their packet forwarding actions (on a probabilistic basis) to an accounting center that consequently remunerates the nodes. The approach we propose here does not rely on reports; instead, we use the concept of session during which each forwarding node authenticates itself to the base station by altering the packet to be forwarded in a specific way. Finally, the protocol proposed in [53] includes routing decisions, whereas the protocols that we propose in this work are independent of routing.

4.8 Conclusion

In this chapter, we proposed a set of protocols that fosters cooperation for the packet forwarding service in Hybrid Ad-hoc networks. Our solution is based on the charging and rewarding of the nodes and relies exclusively on symmetric cryptography to comply with the limited resources of most mobile stations. We have used the concept of sessions, which takes advantage of the relative stability of routes, and we have shown that our scheme stimulates cooperation in Hybrid Ad-hoc networks. Finally, we have analyzed the robustness of our protocols against various attacks and have shown that our solution thwarts rational attacks and detects malicious attacks.

As future work, we intend to consider techniques that aim at the calibration of the relevant parameters, and to study the reaction of the network to sophisticated attacks (e.g., by means of simulations). We will also explore further the statistical detection, at the operator, of malicious attacks and we will study the coexistence of several operators.

Publications:[11, 12]

Chapter 5

Conclusion

In this thesis, we have considered three families of networks; WiFi networks, Wireless Mesh Networks, and Hybrid Ad-hoc networks. For each of these families, we have identified important network functions that require the cooperation of different entities in the network and we have proposed a secure, lightweight and efficient incentive mechanism.

For WiFi networks, we have proposed a solution where the mobile clients cooperatively build a reputation system that encourages the Wireless Internet Service Providers to cooperate (i.e., to provide the clients with a good Quality of Service). Our solution also allows the mobile users to connect to foreign WISPs in a secure way while preserving its anonymity. To the best of our knowledge, our scheme is the only one that is secure, encourages the WISPs to behave well and offers a seamless roaming mechanism, and all for a moderate overhead for the the mobile clients. We have analyzed the robustness of our solution against various attacks and we have shown by means of simulations that our reputation model indeed encourages the WISPs to behave correctly.

For Wireless Mesh networks, we have proposed FAME, a fair scheduling mechanism that optimizes the bandwidth utilization and maximizes the spatial reuse (i.e., the possibility for links that do not contend to be activated at the same time) by assigning transmission rights to the links in the network. With respect to other proposals, FAME is collision-free and it ensures a fair share of the network resources for each client in the network. We have proven that our solution is indeed fair and collision-free, and we have evaluated its efficiency by means of Matlab simulations and by using the Magnets outdoor testbed.

Finally, for Hybrid Ad-hoc networks, we have proposed a charging and rewarding scheme that fosters cooperation for the packet forwarding service. The originality of our solution resides in the fact that it is based on symmetric cryptog-

raphy in order to cope with the limited resources of the mobile stations, that it uses the concept of sessions, which takes advantage of the relative stability of routes, and that it leads to a very moderate overhead. We have analyzed the robustness of our protocols against various attacks and have shown that our solution thwarts rational attacks and detects malicious attacks.

List of Symbols

Symbols Used Throughout the Thesis

$\mathcal{A}$	Attacker
U_X	Utility function of node X
b_X	Benefit received by node X
c_X	Cost of cooperation for node X
$k_{X,Y}$	Symmetric key shared between node X and node Y
$n+1$	Number of elements in a hash chain
$w_i, i = 0..n$	Elements of a hash chain. w_n is the root of the chain and $w_i = h(w_{i+1})$
h	Hash function used to generate the hash chain
N_X	Nonce generated by node X
PK_X	Node X's public key
$Cert(X)$	Certificate of the node X's public key
$E_{PK_X}(M)$	Encryption of message M using node X's public key
$S_{PK_X}(M)$	Signature of message M using node X's public key
$E_{k_{X,Y}}(M)$	Encryption of message M using the symmetric key shared between node X and node Y
$MAC_{k_{X,Y}}(M)$	MAC of message M computed using the symmetric key shared between node X and node Y
ϵ	Small amount of money used in the payment system

Symbols Specific to Chapter 2

H	Home WISP
S	Selected WISP
MN	Mobile Node
TCA	Trusted Central Authority
RR_X	Reputation record of WISP X
AQ_X	QoS advertised by WISP X
RQ_X	The real QoS provided by WISP X
Pr_X	Price per service part required by WISP X
tag	A random number used by H to authenticate MN
C	The contract established between S and MN
α,β, γ	Coefficients used in the decision mechanism

Symbols Specific to Chapter 3

$\mathcal{G}$	Directed graph representing the Mesh Network
$\mathcal{V}$	Set of vertices composed of the TAPs and the WAP
$\mathcal{L}$	Set of links between the TAPs
$\mathcal{L}_U$	Set of upstream links
$\mathcal{L}_D$	Set of downstream links
$\ell_{a,b}$	Link between TAP_a and TAP_b
$C_{\ell_{a,b}}$	Capacity of link $\ell_{a,b}$
n_M	Number of mobile clients
$\mathcal{M}$	Set of mobile clients
f_i^u	Traffic flow generated by the mobile client M_i
f_i^d	Traffic flow received by the mobile client M_i
f_i	Corresponds to f_i^u if we consider upstream traffic and to f_i^d if we consider downstream traffic
T	Duration of the cycle
$t_{\ell_{a,b}}$	Duration of the activation of link $\ell_{a,b}$
$t_{\ell_{a,b}}^{f_i}$	Time dedicated to flow f_i on link $\ell_{a,b}$
$F_{\ell_{i,j}}$	Set of traffic flows traversing link $\ell_{i,j}$

Symbols Specific to Chapter 3 (cnd)

r_i	Route from the TAP serving M_i to the WAP
CM	Compatibility matrix
Cl_c^k	The k-th clique of cardinality c
d_c^k	Time reserved, on the cycle, for Cl_c^k
$g(Cl_c^k)$	Gain associated with the clique Cl_c^k
bl	Bottleneck link
A	Matrix representing the network topology

Symbols Specific to Chapter 4

A	Initiator of the communication
B	Correspondent of the communication
S	Source of a packet (could be A or B)
D	Destination of a packet
BS_A	Base station of the initiator
BS_B	Base station of the correspondent
K_i	Long-term symmetric key shared between node i and the operator
a	Number of intermediate nodes in the initiator route
b	Number of intermediate nodes in the correspondent route
α	Reward for a node that can prove it forwarded a packet
β	Reward for a node that can only prove it received a packet
c	Cost of forwarding a packet
c'	Cost of maintaining and sending the receipt

Bibliography

[1] G. N. Aggélou and R. Tafazolli. On the Relaying Capacity of Next-Generation GSM Cellular Networks. *IEEE Personal Communications*, February 2001.

[2] I. F. Akyildiz, X. Wang, and W. Wang. Wireless Mesh Networks: A Survey. *Computer Networks Journal (Elsevier)*, 47(4), 2005.

[3] R. Anderson and M. Kuhn. Tamper Resistance - a Cautionary Note. In *The Second USENIX Workshop on Electronic Commerce Proceedings*, 1996.

[4] P. Bahl, A. Balachandran, A. Miu, W. Russell, G. Voelker, and Y.M. Wang. PAWNs: Satisfying the Need for Ubiquitous Connectivity and Location Services. *IEEE Personal Communications Magazine (PCS)*, 9(1), 2002.

[5] P. Bahl, A. Balachandran, and S. Venkatachary. Secure Wireless Internet Access in Public Places. In *Proceedings of the IEEE Conference on Communications*, 2001.

[6] F. Bai, N. Sadagopan, and A. Helmy. IMPORTANT: a framework to systematically analyze the Impact of Mobility on Performance of RouTing protocols for Adhoc NeTworks. In *Proceedings of INFOCOM*, 2003.

[7] Y. Bejerano. Efficient Integration of Multi-Hop Wireless and Wired Networks with QoS Constraints. In *Proceedings of Mobicom*, 2002.

[8] Y. Bejerano, S.-J. Han, and L. Li. Fairness and Load Balancing in Wireless LANs Using Association Control. In *Proceedings of MobiCom*, 2004.

[9] M. Bellare and P. Rogaway. Optimal Asymmetric Encryption – How to encrypt with RSA. *Lecture Notes in Computer Science*, 950, 1995.

[10] M. Bellare and P. Rogaway. The Exact Security of Digital Signatures - How to Sign with RSA and Rabin. *Advance in Cryptology - EUROCRYPT*, 1070, 1996.

[11] N. Ben Salem, L. Buttyán, J.-P. Hubaux, and M. Jakobsson. A Charging and Rewarding Scheme for Packet Forwarding in Multi-hop Cellular Networks. In *Proceedings of MobiHoc*, 2003.

[12] N. Ben Salem, L. Buttyán, J.-P. Hubaux, and M. Jakobsson. Node Cooperation in Hybrid Ad-hoc Networks. *IEEE Transactions on Mobile Computing (TMC)*, 5(4), 2006.

[13] N. Ben Salem and J.-P. Hubaux. A Fair Scheduling for Wireless Mesh Networks. In *Proceedings of the IEEE Workshop on Wireless Mesh Networks (WiMesh)*, 2005.

[14] N. Ben Salem and J.-P. Hubaux. Securing Wireless Mesh Networks. *IEEE Wireless Communications Magazine*, 13(2), 2006.

[15] N. Ben Salem, J.-P. Hubaux, and M. Jakobsson. Fuelling WiFi deployment: A reputation-based solution. In *Proceedings of the International Symposium on Modeling and Optimization in Mobile, Ad-Hoc, and Wireless Networks (WiOpt)*, 2004.

[16] N. Ben Salem, J.-P. Hubaux, and M. Jakobsson. Reputation-based Wi-Fi Deployment - Protocols and Security Analysis. In *Proceedings of the ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots (WMash)*, 2004.

[17] N. Ben Salem, J.-P. Hubaux, and M. Jakobsson. Reputation-based Wi-Fi Deployment. *Mobile Computing and Communications Review (MC2R)*, 9(3), 2005.

[18] G. Bianchi. IEEE 802.11Saturation Throughput Analysis. *IEEE Communications Letters*, 2(12), 1998.

[19] P. Bjorklund, P. Varbrand, and D. Yuan. Resource Optimization of Spatial TDMA in Ad-Hoc Radio Networks: A Column Generation Approach. In *Proceedings of INFOCOM*, 2003.

[20] A. Botta, I. Matyasovszki, A. Pescape, and R. Karrer. Performance evaluation of the Magnets backbone. Technical report, Deutsche Telekom Laboratories and University of Napoli Federico II, 2006. http://www.grid.unina.it/Traffic/pub/TR-DTLab-UoN-2006.pdf.

[21] R. Bruno, M. Conti, and E. Gregori. Mesh Networks: Commodity Multihop Ad-Hoc Networks. *IEEE Communications Magazine*, March 2005.

[22] S. Buchegger and J.-Y. Le Boudec. Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes - Fairness In Distributed Ad-Hoc NeTworks. In *Proceedings of MobiHOC*, 2002.

[23] L. Buttyán. Removing the Financial Incentive to Cheat in Micropayment Schemes. *IEE Electronics Letters*, January 2000.

[24] L. Buttyán and J.-P. Hubaux. Stimulating Cooperation in Self-Organizing Mobile Ad-Hoc Networks. *ACM/Kluwer Mobile Networks and Applications (MONET)*, 8(5), 2003.

[25] J. Camp, J. Robinson, C. Steger, and E. Knightly. Measurement Driven Deployment of a TwoTier Urban Mesh Access Network. In *Proceedings of Mobisys*, 2006.

[26] T. Camp, J. Boleng, and V. Davies. A Survey of Mobility Models for Ad-Hoc Network Research. *Wireless Communication and Mobile Computing: Special issue on Mobile Ad-Hoc Networking: Research, Trends and Applications*, 2(5), 2002.

[27] N. Chiba and T. Nashizeki. Arboricity and Subgraph Listing Algorithms. *SIAM J. Comput*, 14, 1985.

[28] D. Coppersmith and M. Jakobsson. Almost Optimal Hash Sequence Traversal. In *Proceedings of Financial Cryptography*, 2002.

[29] M. Crovella and A. Bestavros. Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes. *IEEE ACM Transactions on Networking*, 5(6), 1997.

[30] C. Dellacrocas and P. Resnick. Online Reputation Mechanisms - A Roadmap for Future Research. In *1st Interdisciplinary Symposium on Online Reputation Mechanism*, 2003.

[31] Z. Despotovic and K. Aberer. Trust and Reputation in P2P networks. In *1st Interdisciplinary Symposium on Online Reputation Mechanism*, 2003.

[32] E.C. Efstathiou and G.C. Polyzos. A Peer-to-Peer Approach to Wireless LAN Roaming. In *Proceedings of the ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots (WMash)*, 2003.

[33] T. ElGamal. A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory*, IT-31(4), 1985.

[34] M. Félegyházi, J.-P. Hubaux, and L. Buttyán. Nash Equilibria of Packet Forwarding Strategies in Wireless Ad-Hoc Networks. *IEEE Transactions on Mobile Computing (TMC)*, 5(5), 2006.

[35] V. Gambiroza, B. Sadeghi, and E. Knightly. End-to-End Performance and Fairness in Multihop Wireless Backhaul Networks. In *Proceedings of MobiCom*, 2004.

[36] M. Garey and D. Johnson. *Computers and Intractability - A guide to the Theory of NP-Completeness*. Freeman, San Francisc, 1979.

[37] B. Goode. Voice Over Internet Protocol (VoIP). *Proceedings of the IEEE*, 90, September 2002.

[38] J. Gronkvist. Assignment Methods for Spatial Reuse TDMA. In *Proceedings of MobiHOC*, 2000.

[39] M. Gupta, P. Judge, and M. Ammar. A Reputation System for Peer-to-Peer Networks. In *Proceedings of NOSSDAV*, 2003.

[40] H. Holma and A. Toskala. *WCDMA for UMTS: Radio Access for Third Generation Mobile Communications*. Wiley, 2002.

[41] D. Houser and J. Wooders. Reputation in Auctions: Theory, and Evidence from eBay. Working Paper 00-01, University of Arizona, 2001.

[42] http://www.boingo.com/.

[43] http://www.deutsche-telekom-laboratories.de/∼karrer/magnets.html.

[44] http://www.eskimo.com/˜weidai/benchmarks.html.

[45] http://www.isi.edu/nsnam/ns/.

[46] http://www.rsasecurity.com/products/securid/.

[47] Y.-C. Hu, D. Johnson, and A. Perrig. SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad-Hoc Networks. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, 2002.

[48] Y.-C. Hu and A. Perrig. A Survey of Secure Wireless Ad-Hoc Routing. *IEEE Security and Privacy, special issue on Making Wireless Work*, 2(3), 2004.

[49] Y.-C. Hu, A. Perrig, and D. Johnson. Ariadne: A Secure On-Demand Routing Protocol for Ad-Hoc Networks. In *Proceedings of Mobicom*, 2002.

[50] J.-P. Hubaux, Th. Gross, J.-Y. Le Boudec, and M. Vetterli. Towards Self-Organizing Mobile Ad-Hoc Networks: the Terminodes Project. *IEEE Communications Magazine*, 39(1), 2001.

[51] S. Jain and S. R. Das. Distributed Protocols for Scheduling and Rate Control for MaxMin Fairness in Wireless Mesh Networks. Technical report, State University of New York at Stony brook, 2005.

[52] M. Jakobsson. Financial Instruments in Recommendation Mechanisms. In *Proceedings of Financial Cryptography*, 2002.

[53] M. Jakobsson, J.-P. Hubaux, and L. Buttyán. A Micro-Payment Scheme Encouraging Collaboration in Multi-Hop Cellular Networks. In *Proceedings of Financial Cryptography*, 2003.

[54] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad-Hoc Wireless Networks. In Imielinski and Korth, editors, *Mobile Computing*. Kluwer Academic Publishers, 1996.

[55] R. Karp. Reducibility Among Combinatorial Problems. *Complexity of Computer Computations*, 85-103, 1972.

[56] R. Karrer, I. Matyasovszki, A. Botta, and A. Pescape. Experimental Evaluation and Characterization of the Magnets Wireless Backbone. In *Proceedings of the ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and CHaracterization (WiNTECH)*, 2006.

[57] KaZaA Home Page, http://www.kazaa.com.

[58] Victoria W. Kipp. The battle of NIMBY. PRIMEDIA Business Magazines & Media Inc, 2002.

[59] M. Kodialam and T. Nandagopal. Characterizing the Capacity Region in Multi-Radio Multi-Channel Wireless Mesh Networks. In *Proceedings of MobiCom*, 2005,.

[60] B. Lamparter, K. Paul, and D. Westhoff. Charging Support for Ad-Hoc Stub Networks. *Journal of Computer Communication, Special Issue on Internet Pricing and Charging: Algorithms, Technology and Applications, Elsevier Science*, Summer 2003.

[61] A. Lenstra and E. Verheul. Selecting Cryptographic Key Sizes. *Journal of Cryptology: the JOURNAL of the International Association for Cryptologic Research*, 14(4), 2001.

[62] Y.-D. Lin and Y.-C Hsu. Multihop Cellular: A New Architecture for Wireless Communications. In *Proceedings of INFOCOM*, 2000.

[63] O. Mantel, N. Scully, and A. Mawira. Radio Aspects of Hybrid Wireless Ad-Hoc Networks. In *Proceedings of VTC*, 2001.

[64] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating Routing Misbehavior in Mobile Ad-Hoc Networks. In *Proceedings of Mobicom*, 2000.

[65] A. J. Menezes, P. Van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

[66] P. Michiardi and R. Molva. Core: A Collaborative Reputation Mechanism To Enforce Node Cooperation In Mobile Ad-Hoc Networks. In *Proceedings of CMS*, 2002.

[67] M. Molle and L. Klienrock. Virtual Time CSMA: Why two clocks are better than one. *IEEE transactions on Communications*, 1985.

[68] S. Nelson and L. Kleinrock. Spatial TDMA: A Collision-Free Multihop Channel Access Protocol. *IEEE Transactions on Commnuications*, 33(9), 1985.

[69] P. Papadimitratos and Z. Haas. Secure Routing for Mobile Ad-Hoc Networks. In *Proceedings of CNDS*, 2002.

[70] Boingo Wi-Fi Industry White Paper. Towards Ubiquitous Wireless Broadband. http://www.boingo.com/wi-fi_industry_basics.pdf, 2003.

[71] B. Parno, A. Perrig, and V. Gligor. Distributed Detection of Node Replication Attacks in Sensor Networks. In *Proceedings of IEEE Symposium on Security and Privacy*, 2005.

[72] B. Patel and J. Crowcroft. Ticket based Service Access for the Mobile User. In *Proceedings of MobiCom*, 1997.

[73] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. SPINS: Security Protocols for Sensor Networks. In *Proceedings of Mobicom*, 2001.

[74] P. Resnick and R. Zeckhauser. Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System. In *NBER workshop on empirical studies of electronic commerce*, 2001.

[75] P. Resnick, R. Zeckhauser, J. Swanson, and K. Lockwood. The Value of Reputation on eBay: A Controlled Experiment. In *ESA Conference*, 2002.

[76] R. Rivest and A. Shamir. PayWord and MicroMint: Two simple micropayment schemes. Technical report, MIT Laboratory for Computer Science, 1996.

[77] A. Seshadri, A. Perrig, L. van Doorn, and P. Khosla. SWATT: SoftWare-based ATTestation for Embedded Devices. In *Proceedings of IEEE Symposium on Security and Privacy*, 2004.

[78] L. Tassiulas and S. Sarkar. Maxmin Fair Scheduling in Wireless Networks. In *Proceedings of Infocom*, 2002.

[79] The Emule Project, http://www.emule-project.net.

[80] E. Tomita, A. Tanaka, and H. Takahashi. The Worst-case Time Complexity for Finding all the Cliques. Technical report, UEC-TR-CI, 1988.

[81] P. Varbrand and D. Yuan. Maximal Throughput of Spatial TDMA in Ad-Hoc Networks. In *White paper, Sept.*, 2003.

[82] W. Weiy, K. Suhy, B. Wangz, Y. Guy, and J. Kurosey. Passive Online Rogue Access Point Detection Using Sequential Hypothesis Testing with TCP ACK-Pairs. Technical report, UMass Computer Science Techincal Report, 2006.

[83] H. Wu, C. Qios, S. De, and O. Tonguz. Integrated Cellular and Ad-Hoc Relaying Systems: iCAR. *IEEE Journal on Selected Areas in Communications*, 19(10), 2001.

[84] W. Xu, W. Trappe, Y. Zhang, and T. Wood. The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks. In *Proceedings of MobiHoc*, 2005.

[85] J. Yoon, M. Liu, and B. Noble. Random Waypoint Considered Harmful. In *Proceedings of INFOCOM*, 2003.

[86] A. N. Zadeh, B. Jabbari, R. Pickholtz, and B. Vojcic. Self-Organizing Packet Radio Ad-Hoc Networks with Overlay (SOPRANO). *IEEE Communications Magazine*, June 2002.

[87] M. Zapata and N. Asokan. Securing Ad-Hoc Routing Protocols. In *Proceedings of WiSe*, 2002.

[88] J. Zhang, J. Li, S. Weinstein, and N. Tu. Virtual Operator Based AAA in Wireless LAN Hot Spots with Ad-Hoc Networking Support. *Mobile Computing and Communications Review*, 6(13), 2002.

[89] S. Zhong, Y. R. Yang, and J. Chen. Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks. In *Proceedings of INFOCOM*, 2003.

Index

NAOUEL BEN SALEM

Research and teaching assistant
Laboratory for computer Communications and Applications (LCA)
School of Computer and Communication Sciences (IC)
EPFL (Ecole Polytechnique Fédérale de Lausanne), Switzerland
BC 200, Station 14
CH-1015 Lausanne, Switzerland

e-mail: naouel.bensalem@epfl.ch
url: http://people.epfl.ch/naouel.bensalem
phone: +41 21 6933697

PERSONAL

Born in Tunis, Tunisia on August 19, 1977. Citizen of Tunisia.

RESEARCH

My research focuses on cooperation and security in wireless networks. During my PhD work at EPFL, I have considered different wireless networks and for each of these networks I designed secure incentive mechanisms to foster cooperation.

EDUCATION

PhD. student in communication systems, *Sep. 2001 – present*
EPFL, SWITZERLAND
thesis title: *Secure Incentives to Cooperate for Wireless Networks*
advisor: Prof. Jean-Pierre Hubaux
expected graduation: July 2007

Doctoral school in communication systems, *Oct. 2000 – Jul. 2001*
EPFL, SWITZERLAND

Diploma (Dipl. Ing.) in Computer Science, *Sep. 1997 – Jun. 2000*
ECOLE NATIONALE DES SCIENCES DE L'INFORMATIQUE (ENSI), TUNISIA
Concours National des Ingenieurs, *Sep. 1995 – Jun. 1997*
ECOLE PRÉPARATOIRE AUX ETUDES D'INGENIEURS DE TUNIS (IPEIT), TUNISIA

PROFESSIONAL EXPERIENCE

TEACHING

Supervised projects

Gunnar Schaefer, *Assessing Fairness in Wireless Mesh Networks*, course, 2007
Nicolas Forel, *A Fair Scheduling for Wireless Mesh Networks*, course, 2006
Gildas Avoine, *Cryptanalysis of a Micro-Payment Scheme designed for Asymmetric Multi-Hop Cellular Networks*, course, 2005
Karim Mardam Bey and Amine Soufiane Zahiri, *WiFi networks - Implementation and real-life experiment*, semester, 2004
Lianick Houmgny, *Simulating a multi-hop cellular network using ns-2*, semester, 2004
Thomas Schmid, *Reputation-based WiFi deployment: ns-2 Implementation and analysis* semester, 2004
Martin Rubli, *A multi-hop cellular network simulator*, semester, 2004
Thomas Seiler, *A simulator dedicated to multi-hop cellular networks*, semester, 2003

Teaching assistant

Mobile Networks, EPFL – 2005, 2006, 2007
Self-Organized Mobile Networks, EPFL – 2003, 2004, 2005, 2006, 2007
Computer Networking I, EPFL – 2005
Object Oriented Programming - JAVA, EPFL – 2001

PROFESSIONAL ACTIVITIES

Research and teaching assistant, *Sep. 2001 – present*
SCHOOL OF COMPUTER AND COMMUNICATION SCIENCES (IC)
EPFL, SWITZERLAND

Reviewer for scientific journals

IEEE Transactions on Mobile Communications (TMC)
IEEE Journal on Selected Areas in Communications Multi-hop Wireless Mesh Networks (JSAC)
IEEE Wireless Communications
ACM/Kluwer Mobile Networks and Applications (MONET)
Elsevier Ad Hoc Networks Journal (ADHOC)
International Journal of Computer Systems Science and Engineering (IJCSSE)

Reviewer for scientific conferences, workshops

ACM Mobicom, ACM Mobihoc, IEEE Infocom, IEEE WiOpt, ACM WiSe, ACM SenSys, ACM SigComm, ACM VANET, IEEE TSPUC

AWARDS

EPFL doctoral school fellowship, Oct. 2000 – Jul. 2001
3rd price at ENSI, Tunisia, Jun. 2000
2nd price at ENSI, Tunisia, Jun. 1999
1st price at ENSI, Tunisia, Jun. 1998

GENERAL IT SKILLS

Programming: C, C++, HTML, Ns-2, Java, Pascal, Ada,

Operating systems: Windows, Linux, Unix, DOS

Mathematical tools: MatLab

Presentation: LaTeX, Microsoft Word, Powerpoint, Excel

REFERENCES

Prof. Jean-Pierre Hubaux, Full Professor,
EPFL, Switzerland
Room BC 207, +41 21 6932627, jean-pierre.hubaux@epfl.ch

Prof. Markus Jakobsson, Associate Professor,
INDIANA UNIVERSITY, BLOOMINGTON, USA
Eigenmann 1026, (812) 856 1807, markus@indiana.edu

PUBLICATIONS

Journals

1. N. Ben Salem and J.-P. Hubaux, **Securing Wireless Mesh Networks**, IEEE Wireless Communications, vol. 13, nr. 2, 2006.
2. N. Ben Salem, L. Buttyán, J.-P. Hubaux, and M. Jakobsson, **Node Cooperation in Hybrid Ad hoc Networks**, IEEE Transactions on Mobile Computing (TMC), vol. 5, nr. 4, 2006.
3. N. Ben Salem, J.-P. Hubaux, and M. Jakobsson, **Reputation-based Wi-Fi Deployment**, ACM Mobile Computing and Communications Review (MC2R), vol. 9, nr. 3, 2005.

Conferences, workshops

4. N. Ben Salem and J.-P. Hubaux, **A Fair Scheduling for Wireless Mesh Networks**, in Proceedings of the IEEE Workshop on Wireless Mesh Networks (WiMesh), Santa Clara, CA, USA, September 2005.
5. N. Ben Salem, J.-P. Hubaux, and M. Jakobsson, **Reputation-based Wi-Fi Deployment: Protocols and Security Analysis**, in Proceedings of the ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots (WMASH), Philadelphia, USA, October 2004.
6. N. Ben Salem, J.-P. Hubaux, and M. Jakobsson, **Fuelling WiFi deployment: A reputation-based solution**, in Proceedings of the International Symposium on Modeling and Optimization in Mobile, Ad-Hoc, and Wireless Networks (WiOpt), Cambridge, UK, March 2004.
7. N. Ben Salem, L. Buttyán, J.-P. Hubaux, and M. Jakobsson, **A Charging and Rewarding Scheme for Packet Forwarding in Multi-hop Cellular Networks**, in Proceedings of MobiHoc, Annapolis, USA. June, 2003.
8. L. Buttyán and N. Ben Salem, **A Payment Scheme for Broadcast Multimedia Streams**, in Proceedings of ISCC, Hammamet, Tunisia, 3-5 July 2001.

Technical reports, posters

9. N. Ben Salem, J.-P. Hubaux, and M. Jakobsson, **Reputation-based Wi-Fi Deployment: Protocols and Security Analysis**, EPFL Technical Report LCA-REPORT-2004-014, 2004.
10. N. Ben Salem, L. Buttyán, J.-P. Hubaux, and M. Jakobsson, **Cooperation in Multi-hop Cellular Networks**, EPFL Technical Report LCA-REPORT-2003-015, 2003.
11. N. Ben Salem, L. Buttyán, J.-P. Hubaux, and M. Jakobsson, **Incentive mechanisms in multi-hop wireless networks**, Poster at the International Symposium on Modeling and Optimization in Mobile, Ad-Hoc, and Wireless Networks (WiOpt), INRIA Sophia-Antipolis, France. March 3-5, 2003.

Under submission

12. N. Ben Salem, R. Karrer, A. Feldmann and J.-P. Hubaux, **Fairness in Wireless Mesh Networks**, submitted to IEEE Transactions on Wireless Communications (TWC).

Lausanne, Mar. 27, 2007